

Informační systém poptávkového portálu

Information System for Electronic Inquires

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostravě 16. dubna 2009

.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 16. dubna 2009

.....

Rád bych na tomto místě poděkoval všem, kteří mi s prací pomohli, protože bez nich by tato práce nevznikla.

Abstrakt

Tato práce se zabývá vývojem kompletního informačního systému poptávkového portálu. První část práce se zaměřuje na přehled technologií, které byly při implementaci této práce využity. Dále se práce věnuje analýze poptávkových portálů. Součástí jsou také základní požadavky na nově vyvíjený systém a inovace. Další část práce se zabývá specifikací požadavků, které jsou na systém kladeny. Část analýza a návrh popisuje statickou a dynamickou strukturu systému, včetně návrhu nasazení. Poslední část práce se věnuje vlastní implementaci systému, architekturou a nasazením systému, který je schopen komerčního využití.

Klíčová slova: poptávkový portál, ASP.NET, AJAX, Silverlight, LINQ to SQL, dodavatel, odběratel, poptávka

Abstract

This work deals with the development of a complete information system for electronic inquiries. The first part focuses on an overview of technologies that were used by the implementation of this work. Further work concentrates on the analysis of demand portals. It also includes the basic requirements for newly developed systems and innovation. Next part deals with the specification of requirements that are placed on the system. The part of the work containing the analysis and design describes the static and dynamic structure of the system, including the proposal of deployment. The last part deals with the implementation of the system, architecture and deployment system that is capable of commercial use.

Keywords: demand portal, ASP.NET, AJAX, Silverlight, LINQ to SQL, contractor, subscriber, demand

Seznam použitých zkratk a symbolů

ADO.NET	– ActiveX Data Objects for .NET
AJAX	– Asynchronous JavaScript and XML
ASP	– Active Server Pages
ASP.NET	– Active Server Pages for .NET
CLR	– Common Language Runtime
CMS	– Content management system
CSS	– Cascading Style Sheets
HTML	– HyperText Markup Language
HTTP	– Hypertext Transfer Protocol
HTTPS	– Hypertext Transfer Protocol Secure
IDE	– Integrated development environment
IIS	– Internet Information Services
IS	– Informační systém
LINQ	– Language INtegrated Query
MVC	– Model View Controller
ORM	– Object-relational mapping
PDF	– Portable Document Format
PHP	– Hypertext Preprocessor
PIN	– Personal Identification Number
RTF	– Rich Text Format
SEO	– Search Engine Optimization
SQL	– Structured Query Language
SSL	– Secure Sockets Layer
TIFF	– Tagged Image File
UML	– Unified Modeling Language
URL	– Uniform Resource Locator
VISA	– Visa International Service Association
WYSIWYG	– What You See Is What You Get
XAML	– Extensible Application Markup Language
XHTML	– Extensible HyperText Markup Language
XML	– Extensible Markup Language
XSLT	– Extensible Stylesheet Language Transformations

Obsah

1	Úvod	6
2	Použité technologie	7
2.1	ASP.NET	7
2.2	LINQ	9
2.3	ASP.NET AJAX	11
2.4	SILVERLIGHT	11
3	Analýza poptávkových portálů	13
3.1	Jak funguje poptávkový portál?	13
3.2	Vysvětlení pojmů	13
3.3	Přehled poptávkových portálů	15
3.4	Specifikace nového systému	20
3.5	Inovace	20
4	Specifikace požadavků	22
4.1	Identifikace funkčních požadavků	22
4.2	Specifikace případů užití	25
4.3	Nefunkční požadavky	38
5	Analýza a návrh	39
5.1	Analýza	39
5.2	Návrh	39
6	Implementace	47
6.1	Architektura systému	47
6.2	Bezpečnost aplikace	48
6.3	SEO - optimalizace pro vyhledávače	49
6.4	URL rewriting	51
6.5	Tag cloud	52
6.6	Silverlight a statistiky	53
6.7	Elektronický platební systém	54
6.8	Exporty	55
7	Závěr	57
7.1	Další vývoj projektu	57
8	Literatura	59
	Přílohy	59

A	Uživatelské rozhraní a ukázky aplikace	60
A.1	Uživatelské rozhraní	60
A.2	Ukázky aplikace	60
B	Datový slovník	65

Seznam tabulek

1	Tarify, nabízené portálem Epopstavka.cz	16
2	Tarify, nabízené portálem AAA poptávka	18
3	Balíčky služeb, nabízené portálem AAA poptávka	18
4	Priority požadavků	22
5	Funkční požadavky - Zákazníci	25
6	Funkční požadavky - Uživatelé	26
7	Funkční požadavky - Poptávky	26
8	Funkční požadavky - Nabídky	27
9	Funkční požadavky - Zprávy	27
10	Funkční požadavky - Pobočky	27
11	Funkční požadavky - Tarify	28
12	Funkční požadavky - CMS	28
13	Funkční požadavky - Analytiky	28
14	Funkční požadavky - Mailing	28
15	Funkční požadavky - Faktury - platby	29
16	Datový slovník - CMS_actuality	65
17	Datový slovník - CMS_banner	65
18	Datový slovník - CMS_article	66
19	Datový slovník - CMS_category	66
20	Datový slovník - CMS_section	66
21	Datový slovník - CMS_menu	67
22	Datový slovník - CMS_menu_item	67
23	Datový slovník - CMS_type	67

Seznam obrázků

1	Epoptavka.cz – ukázka portálu	17
2	AAA poptávka – ukázka portálu	19
3	Use Case diagram – Kontext	30
4	Use Case diagram – Systém	31
5	Use Case diagram – Administrační systém	33
6	Use Case diagram – CMS	35
7	Use Case diagram – Zadání poptávky	36
8	Třídní diagram analytický	40
9	Návrh databáze	41
10	Třídní diagram – CMS	42
11	Sekvenční diagram – Odeslání poptávky	44
12	Stavový diagram - Verifikace zákazníka	45
13	Diagram nasazení	45
14	Architektura systému	48
15	Tag Cloud – ukázka	52
16	Statistika dodavatelů a odběratelů	54
17	GP WebPay - 3-D Secure	55
18	Uživatelské rozhraní	61
19	Prezentace dat pomocí komponenty RadGrid	62
20	Okno pro odeslání interní zprávy	63
21	Export faktury ve formátu PDF	64

Seznam výpisů zdrojového kódu

1	LINQ to SQL - datový kontext	10
2	LINQ to SQL - ukázka získávání dat	10
3	LINQ to SQL - dotaz pomocí rozšiřujících metod a lambda výrazů	10
4	LINQ to SQL - ukázka vkládání dat	10
5	LINQ to SQL - ukázka úpravy dat	11
6	LINQ to SQL - ukázka mazání dat	11
7	Příklad implementace metody v datové vrstvě	47

1 Úvod

V globálním měřítku se dnes projevuje dopad celosvětové krize. Zasáhla malé i velké společnosti. Ty se snaží získat nové zakázky, či nabízet své služby mnohem razantněji, než tomu bylo v předešlých letech. Zvyšující se konkurence nutí firmy hledat nové prostředky, jak být na trhu úspěšnější. Mocnou zbraní je samozřejmě Internet. Je jednou z cest, jak svoji firmu zviditelnit. Proto jsou firmy ochotny investovat nemalé peníze za reklamu, zřizují si nové prezentační stránky a oslovují odborníky v oblasti SEO, aby právě oni byly co nejvýše ve vyhledávačích. A proč vlastně tolik vynaloženého úsilí a finančních prostředků? Jde o způsob, jak dát okolnímu světu o firmě vědět, jak získat nové zákazníky, jak získat nová odbytiště. Bohužel je to cesta zdlouhavá.

Již několik let se však na internetu objevuje jiné řešení - Poptávkové portály. Pokud se firma rozhodne využít služeb tohoto druhu portálu, nebude již nucena čekat na zákazníky. Tyto systémy umožňují firmám kontaktovat zákazníky a přímo jim nabídnout služby, které by uspokojili, jejich poptávky. Stejně tak může firma prezentovat svou činnost a může být oslovena zákazníkem.

Na českém trhu mnoho kvalitních portálů nenajdete. Většina z nich již zaostává za trendy a novými technologiemi dnešního internetu a proto se vyskytuje v této oblasti prázdné místo. To si uvědomila i společnost XEVOS Solutions s.r.o., která se zaměřuje především na vývoj obchodně orientovaných aplikací s využitím nástrojů Business Intelligence. Rozhodla se, že si nechá zhotovit specifický poptávkový portál, díky kterému by prázdné místo zaplnila kvalitním produktem, který by poskytoval odpovídající služby. Má diplomová práce se tedy zabývá vývojem takového poptávkového portálu. Práce je členěna do několika kapitol.

Kapitola 2 - Použité technologie - kapitola popisuje hlavní technologie, které byly při vývoji systému použity.

Kapitola 3 - Analýza poptávkových portálů - vysvětlení základních pojmů, přehled českých poptávkových portálů a služeb, které poskytují. Díky tomuto přehledu budeme schopni využít získané informace, při specifikování vlastního poptávkového portálu. V této kapitole jsou také popsány inovace, které bude nový systém obsahovat.

Kapitola 4 - Specifikace požadavků - kapitola popisuje celý proces získávání požadavků od zadavatele a jejich bližší specifikování pomocí use-case diagramů.

Kapitola 5 - Analýza a návrh - tato kapitola ukazuje, jakým způsobem budeme systém realizovat v implementační fázi. Pomocí UML diagramů budeme analyzovat a navrhovat statickou a dynamickou strukturu systému.

Kapitola 6 - Implementace - popisuje architekturu vyvíjeného systému a blíže specifikuje prvky, které byly v systému implementovány.

2 Použité technologie

2.1 ASP.NET

ASP.NET je webový aplikační framework vytvořen společností Microsoft. ASP.NET umožňuje programátorům vytvářet dynamické webové stránky, webové aplikace a webové služby. Je přímým nástupcem technologie ASP, který je také produktem společnosti Microsoft a který byl navržen jako sada nástrojů pro vkládání dynamického obsahu na běžné webové stránky. ASP.NET je postaven na Common Language Runtime (CLR), což umožňuje programátorům psát kód v několika podporovaných jazycích .NET, jako jsou například C#, či Visual Basic. Navíc sdružuje několik nových technologií, které přináší kompletní revoluci, od přístupu do databáze až po distribuované aplikace. Je jednou z nejdůležitějších komponent .NET Framework - jedná se o část, která umožňuje vyvíjet velmi výkonné webové aplikace. Bez nadsázky můžeme říci, že ASP.NET představuje nejkompletnější platformu pro vývoj webu, která kdy byla sestavena. Daleko převyšuje svého předchůdce ASP. Představuje plnohodnotnou platformu pro vývoj komplexních a velmi rychlých webových aplikací [1].

2.1.1 ASP.NET WebForms vs. ASP.NET MVC

V současné době Microsoft nabízí dvě oddělené vývojové větve framework, které jsou a budou i nadále vyvíjeny samostatně. Jedná se o ASP.NET WebForms a ASP.NET MVC. Je třeba zvolit ten, který bude pro naše účely nejvhodnější. Proto si obě technologie blíže popíšeme a na základě získaných informací rozhodneme, který framework pro vývoj systému využijeme.

2.1.1.1 ASP.NET WebForms Počínaje .NET Framework verzí 1.0 v lednu 2002 byl Web Forms prvním opravdovým pokusem společnosti Microsoft, nabídnout nový nástroj pro tvorbu webových aplikací. Prioritou byla robustnost, flexibilita a musel také vyhovět požadavkům webu té doby. ASP.NET WebForms prokázal, že vyspělou technologií bezesporu je. Běží na ní stejně dobře malé i velice rozsáhlé weby. WebForms byla vyvinuta na bázi Windows Form, která vychází z deklarativního programování a využívá událostí řízený model, tzv. event-driven model. Podobnost s Windows Form designérům dovovala, aby využili plně výhody funkčnosti drag-and-drop (česky táhni-a-pust'), WYSIWYG editoru, rozhraní, které se stalo zvykem při vývoji Windows Form ve Visual Studiu 6.0.

Výhody:

- obsahuje sadu tzv. chytrých ovládacích prvků - je možné použít spoustu hotových knihoven
- vyspělá technologie
- jednoduchý vývoj rozsáhlých webových aplikací
- podobnost vývoje desktopových aplikací

- WebForms je skvělá volba pro vývoj silně datově řízených aplikací
- výborná podpora WYSIWYG editoru ve Visual Studiu

Stinné stránky:

- prezentační vrstva obsahuje i část logiky - ta je obsažena v tzv. code-behind souborech
- je těžké provádět automatické testování aplikací
- aby se dosáhlo bezstavosti, je stav každé stránky uložen ve ViewState, který je interpretován na klientské straně jako skryté pole. To vede k mnohdy velkým velikostem stránky.

2.1.1.2 ASP.NET MVC Microsoft na podzim roku 2007 oznámil, že vytváří platformu založenou na jádře ASP.NET, která bude konkurovat jiným populárním MVC platformám. Microsoft vytvořil ASP.NET MVC, což je moderní webová platforma založená na návrhovém vzoru Model-View-Controller. Tento návrhový vzor dělí aplikaci do vrstev, čímž poměrně striktně odděluje funkční, doménovou a prezentační logiku. Navíc dává vývojářům plnou kontrolu nad prezentační vrstvou, tedy nad výstupem. To je hlavní a nejdůležitější rozdíl mezi WebForms a MVC.

Výhody:

- poskytuje kontrolu nad výstupem, proto čistší HTML
- poskytuje unit testování na aplikační vrstvě
- podporuje četné zobrazovací nástroje, jako je Brail, NHaml, NVelocity, XSLT, atd.
- jednoduchá integrace JavaScriptových framework jako je jQuery
- URL je mapována logicky a dynamicky, závislé na uživateli
- bez ViewState aPostBack modelů
- podporuje některé prvky ASP.NET, jako jsou autentifikace, cachování, atd.
- velikost stránek generované MVC jsou většinou mnohem menší, kvůli vyřazení ViewState

Stinné stránky:

- vyžaduje znalost přinejmenším HTML, CSS a JavaScriptu
- dosud slabá podpora open-source i komerčních knihoven

2.1.1.3 Výběr vhodného přístupu Jak bylo v předchozích bodech ukázáno, mají ASP.NET WebForms a MVC své výhody i nevýhody. Shrnutím by se dalo říct, že pokud chceme mít absolutní kontrolu nad prezentační vrstvou, chceme se sami starat o webové standardy nebo o přístupnost, můžeme zvolit MVC model. Pokud chceme používat chytřích ovládacích prvků a dáváme přednost událostí-řízenému webovému vývoji, budeme volit spíše WebForms model. Po zvážení mnoha aspektů, bylo nakonec rozhodnuto o využití WebForms modelu. Na rozhodnutí měla vliv také osobní zkušenost s tímto modelem a v neposlední řadě také plánované využití vyspělých komponent od společnosti Telerik, právě pro WebForms.

2.2 LINQ

Jedná se o nástroj ASP.NET pro získávání a modifikaci dat. LINQ aplikuje principy objektově orientovaného programování na relační data. To představuje jednotný programovací model pro dotazování dat z různých typů datových zdrojů a rozšiřuje přímo funkcionalitu jazyků C# a Visual Basic [5].

LINQ dovoluje dotazování, přidávání, filtrování, mazání a editaci dat v polích, kolekcích odvozených od generického rozhraní `IEnumerable<T>`, XML, či SQL databázích. Dohromady tato technologie nabízí čtyři hlavní implementace:

- LINQ to Objects - Manipulace s daty v paměti
- LINQ to SQL - Manipulace s daty v databázích hostovaných na MS SQL server
- LINQ to DataSet - Manipulace s daty uloženými v ADO.NET datasetech
- LINQ to XML - Manipulace s daty v XML formátu

Nejpopulárnější je beze sporu LINQ to SQL, který se používá pro komunikaci s databází Microsoft SQL Server. Jedná se o ORM (objektově-relační mapper), díky čemuž můžeme snadno namapovat tabulky v databázi na standardní .NET třídy, kde jednotlivé veřejné vlastnosti odpovídají jednotlivým sloupcům tabulky. Objekty dané třídy pak reprezentují řádky tabulky. LINQ to SQL plně podporuje transakce, pohledy, a uložené procedury. Také představuje jednoduchou cestu, jak integrovat validaci dat a pravidla business logiky do datového modelu. LINQ to SQL má velice jednoduché použití a plně nahrazuje psaní klasických SQL dotazů.

Než začneme přímo využívat LINQ to SQL musíme provést namapování tabulek. Jediné, co musí uživatel ve Visual Studiu 2008 udělat, je vybrat tabulky z databáze a doslova je přetáhnout do design modu. Odpovídající třídy se nám samy vygenerují, včetně vztahů, které mezi nimi existují. Pak můžeme začít psát LINQ dotazy. Jako příklad použití, ukážeme základní operace, se kterými LINQ to SQL pracuje.

2.2.1 Komunikace s databází

Třída *DataContext*, představuje hlavní cestu pro komunikaci s databází. Pokud tedy budeme mapovat tabulky ve Visual Studiu v design modu určeném pro LINQ to SQL,

vytvoří se po prvním uložení automaticky třída, která bude dědit ze třídy *DataContext* a která bude reprezentovat tabulky, případně uložené procedury, které jsme při mapování použili. Pokud bychom například databázi nazvali *NameOfDatabase*, bude se v tomto případě naše třída pro přístup k databázi jmenovat *NameOfDatabaseDataContext*. Vše se vytvoří automaticky a lze samozřejmě posléze modifikovat.

```
NameOfDatabaseDataContext db = new NameOfDatabaseDataContext();
```

Výpis 1: LINQ to SQL - datový kontext

Nyní ukážeme základní operace, které nástroj LINQ to SQL poskytuje. V příkladech budeme využívat fiktivní tabulku *Product*, která bude představovat soubor produktů. Tato tabulka bude mít tři atributy. Identifikační číslo záznamu (*Id*), cenu (*Price*) a název produktu (*ProductName*).

2.2.2 Získávání dat

Pokud chceme například vybrat všechny produkty, jejichž cena je větší než 2000, použili bychom následující kód:

```
var products = from p in db.Products where p.Price > 2000 select p;
```

Výpis 2: LINQ to SQL - ukázka získávání dat

LINQ je možné používat i v méně upovídané formě také pomocí rozšiřujících metod a lambda výrazů, které slouží ke zkrácení dotazu. Tento kód má stejnou funkcionalitu jako předchozí dotaz.

```
var products = db.Products.Where(p => p.Price > 2000).Select(p => p);
```

Výpis 3: LINQ to SQL - dotaz pomocí rozšiřujících metod a lambda výrazů

2.2.3 Vkládání dat

Kód níže přidá do databáze nový produkt. Žádné změny v databázi nejsou uloženy, dokud explicitně nezavoláme metodu *SubmitChanges*. LINQ to SQL balí automaticky všechny změny do databázové transakce, což je označení pro atomickou (nedělitelnou) akci, takže buďto se nakonec uloží všechny změny, nebo žádné, nic mezi tím.

```
Product newProduct = new Product();
newProduct.ProductName = "MacBook_Pro";
newProduct.Price = 35000;
db.Products.InsertOnSubmit(newProduct);
db.SubmitChanges();
```

Výpis 4: LINQ to SQL - ukázka vkládání dat

2.2.4 Upravování dat

Pokud chceme upravit některý ze záznamů, stačí jen vybrat odpovídající položku, změnit hodnotu atributů a zavolat metodu *SubmitChanges* pro uložení změny. Následující kód vybere první záznam z tabulky *Product*, jehož název začíná řetězcem "MacBook", změní atribut *Price* a potvrdí změny.

```
Product product = db.Products.First(p => p.ProductName.StartsWith("MacBook"));
product.Price = 33000;
db.SubmitChanges();
```

Výpis 5: LINQ to SQL - ukázka úpravy dat

2.2.5 Mazání dat

Mazání je stejně jednoduché jako jiné operace. Jako příklad si můžeme ukázat výběr produktu, který následně smažeme. Vybereme tedy produkt, jehož atribut *Id* je 100 a pomocí metody *DeleteOnSubmit* určíme, že se má záznam smazat z tabulky *Product*. Vše je samozřejmě nutné potvrdit metodou *SubmitChanges*.

```
Product product = db.Products.Single(p => p.Id == 100);
db.Products.DeleteOnSubmit(product);
db.SubmitChanges();
```

Výpis 6: LINQ to SQL - ukázka mazání dat

2.3 ASP.NET AJAX

Jde o AJAX framework pro tvorbu interaktivních webových aplikací, které pracují napříč všemi populárními prohlížeči. ASP.NET AJAX umožňuje vývojářům vybrat si jejich preferovanou metodu, ať je to programování na straně serveru, klienta, nebo jejich kombinace. Protože jde o velice oblíbenou technologii, existují na internetu mnohé řešení a hotové komponenty. Microsoft dokonce poskytuje balík komponent Ajax Control Toolkit, který je volně ke stažení na oficiálních stránkách. I samotné použití AJAX je velice jednoduché. Stačí do ASPX stránky vložit komponentu *ScriptManager*, která zajistí připojení potřebných AJAX JavaScriptů, jež zajišťují obsluhu objektů na straně klienta. Poté můžeme začít využívat široké možnosti této technologie. V této práci bude využito komponent již zmiňovaného balíku Ajax Control Toolkit a také sady AJAX komponent od společnosti Telerik.

2.4 SILVERLIGHT

Silverlight je framework pro tvorbu webových aplikací, který je svou funkcionalitou podobný technologii Adobe Flash. Podobně jako Flash umožňuje i Silverlight vytvářet interaktivní obsah, který běží u klienta ve společném běhovém prostředí, a podporuje

multimédia, dynamické grafiky a animace, které daleko přesahují možnosti obyčejného HTML. V aplikacích založených na této technologii je uživatelské rozhraní deklarováno v jazyku XAML a k programování logiky se používá zredukované verze společných runtime jazyků (common language runtime, CLR) .NET, a umožňuje proto vývojářům psát kód u klienta v čistém C#, či jazyku Visual Basic. Silverlight je kompatibilní s většinou webových prohlížečů používaných na operačních systémech Microsoft Windows, Linux a Mac OS X [6].

Výhody:

- Vývojové prostředí - defaultním prostředím pro vývoj Silverlight aplikací je Visual Studio ve verzi 2008 a 2010. Je tedy jedno zdali programujete v C, C++, C#, či Javě, můžete vytvářet aplikace stále ve stejném IDE, aniž byste museli kupovat nový produkt a učit se jej.
- .NET Framework a podpora .NET jazyků - Silverlight běží pod .NET Framework. Proto je možné použít jazyků C# a Visual Basic. Silverlight také podporuje psaní aplikací v dynamických jazycích jako je Iron Ruby.
- Integrace s ostatními .NET technologiemi - můžeme využívat velké množství technologií, na které jsme byly doposud zvyklí. Jako dobrý příklad je využití technologie LINQ. Díky tomu můžeme pracovat s datovými strukturami rychle a efektivně.

3 Analýza poptávkových portálů

3.1 Jak funguje poptávkový portál?

Poptávkové portály fungují na velice běžném principu poptávky - nabídky. Proces poptávání a nabízení je díky internetu velice rychlý a proto sebou nese mnoho výhod, jak z pohledu poptávajícího, tak i z pohledu dodavatele služby. Uživatelé totiž mají na jednom místě soustředěny poptávky a nabídky ze všech možných oblastí produktů a služeb.

3.1.1 Z pohledu poptávajícího

Pokud jako poptávající potřebujete například firmu, která by vám postavila dům, zadáte do některého poptávkového portálu svou poptávku. Můžete samozřejmě určit různá kritéria, jako jsou cena, za kterou chcete dům postavit, či jiné požadavky, na kterých trváte. Pak už jen čekáte, až se vám ozve některá ze stavebních firem s nabídkou. Vy posléze vyberete, která nabídka je pro vás ta nejvýhodnější. Poptávající si jednoduše cení toho, že nabídky jsou okamžitou a přímou reakcí na jejich poptávku.

3.1.2 Z pohledu dodavatele

Dodavatel, tedy firma či soukromá osoba, se zaregistruje do katalogu firem. Poté může procházet databází poptávek a reagovat na ně svými nabídkami. Má tak možnost prezentovat svou činnost a zároveň může oslovovat zákazníky cíleně dle jejich poptávky, čímž může získat snadno a rychle nové zakázky.

3.2 Vysvětlení pojmů

3.2.1 Dodavatel

Dodavatel je firma, která má přístup k poptávkám soukromých osob, firem i státních institucí. Aby mohl dodavatel získávat poptávky a reagovat na ně, musí se zaregistrovat. Základní registrace probíhá na většině poptávkových portálů zdarma. Rozdíl mezi registrací zdarma a některým z nabízených zpoplatněných tarifů, bývá většinou v časovém omezení přístupu ke kontaktu na poptávajícího. Pokud tedy zaplatíte příslušný poplatek, který se pohybuje kolem několika tisíc ročně, máte přístup ke kontaktům okamžitě, a tedy můžete být mezi prvními, kdo poptávajícího osloví. Na rozdíl od uživatelů s neplacenou registrací. Těm je zpřístupněn kontakt až po určitém časovém období (např. týdenní prodleva), či dokonce nemusí mít ke kontaktům přístup vůbec. Dále je dle tarifu rozlišen i rozsah prezentace firmy. U nejdražších služeb dávají portály firmě možnost, vkládat své fotky, prezentační videa, podrobné informace a jiné. U těch základnějších například strohé informace o firmě a logo. Žádnou provizi z realizované zakázky ani jiné další poplatky již dodavatel většinou platit nemusí.

Výhody poptávkových portálů pro dodavatele:

- Dodavatel je zapsán do katalogu firem, ve kterém může prezentovat svou firmu a zde si jej můžou poptávající najít a kontaktovat přímo.
- Poptávky, o které má dodavatel zájem, mu jsou zasílány e-mailem. Oborů může být zvoleno několik.
- Většinou jsou k dispozici i historie, analytiky (záleží na poskytovateli).

3.2.2 Odběratel (poptávající)

Odběratel je soukromá osoba, nebo firma, která poptává jakékoliv zboží, či služby, z jakéhokoliv oboru. Poptávku uživatel zadá vyplněním jednoduchého formuláře, kde určí, co poptává, a vyplní kontaktní údaje. Poptávkový portál se již postará, o kontaktování všech dodavatelů, kteří mají o dané odvětví poptávek zájem. Ti posléze prostřednictvím e-mailu, či telefonicky kontaktují poptávajícího. Z řady nabídek si pak poptávající může vybrat tu, která mu nejlépe vyhovuje, ať už cenou, zbožím nebo termínem dodání. Vše funguje zdarma a bez nutnosti registrace. Některé servery však registraci přece jen nabízejí. Uživatel má po přihlášení možnost zadávat poptávky bez vyplňování kontaktních údajů, nebo sledovat poptávky minulé. Některé servery dokonce neregistrovaným uživatelům nedovolují procházet katalog firem nebo pouze s omezeným přístupem.

Výhody poptávkových portálů pro odběratele:

- Možnost vkládání poptávek, selekce nabídek.
- Získání nabídek od řady dodavatelů, bez nutnosti oslovovat jednotlivé dodavatele.
- V katalogu firem můžete procházet firemní prezentace a kontaktovat je s vaší poptávkou přímo.

3.2.3 Poptávky

Poptávky tvoří databázi, která je přístupná všem návštěvníkům portálu. Jsou řazeny podle oborů. Návštěvník portálu či registrovaná firma tak může procházet a hledat poptávky a sledovat její obsah. Avšak pokud nejste registrován jako dodavatel, nebude vám zobrazen kontakt na poptávajícího. Tím jsou možnosti neregistrovaných značně omezeny.

3.2.4 Veřejné poptávky (zakázky)

Některé portály nabízejí také vkládání veřejných zakázek. Ty jsou určeny pro všechny subjekty veřejné správy (např. obecní úřady, školy, kulturní zařízení, zdravotní zařízení), které musejí nakupovat zboží či služby v souladu se zákonem o veřejných zakázkách. Veřejné zakázky se řídí zákonem č. 137/2006 Sb., o veřejných zakázkách [7].

Základním bodem pro získání některé z veřejných zakázek je nutnost splňovat podmínky účasti ve výběrovém řízení. Podmínky účasti ve výběrovém řízení jsou:

1. kvalifikační předpoklady dodavatelů, včetně požadavků týkajících se registrace v profesních a obchodních rejstřících
2. technická, ekonomická a finanční způsobilost

Zadavatel poté vybere z přijatých nabídek a napíše písemné rozhodnutí a pořadí uchazečů. Nabídka je posuzována podle hodnotících kritérií. V nejčastějších případech je rozhodujícím kritériem nejnižší nabídková cena, ale i záruka a servisní podmínky bývají častým kritériem zadavatele.

3.3 Přehled poptávkových portálů

V současnosti se na internetu vyskytuje hned několik poptávkových portálů. Dosud však nebylo vypracováno žádné srovnání, které by tyto portály blíže zkoumalo. Proto se o to pokusíme sami. Cílem ovšem není zjistit, který z poptávkových portálů je tím nejlepším. Jedná se o přehled, co poptávkové portály uživatelům nabízejí, a o analyzování faktorů, které mohou být při výběru portálu pro potenciálního dodavatele, či odběratele důležité. Tyto poznatky poté můžeme využít při tvorbě našeho systému a následně je inovovat.

Na poptávkové portály můžeme pohlížet dle několika kritérií. Tyto kritéria vycházejí z požadavků potenciálních uživatelů. Mezi ty, které mohou být pro volbu poptávkového portálu klíčové, můžeme zařadit zaměření portálu dle oboru, cenu služeb, množství poptávek a nabídek, škálu nabízených funkcí, uživatelskou podporu a v neposlední řadě uživatelskou přívětivost portálu.

- **zaměření portálu** - některé poptávkové portály mohou být všeobecné, jiné mohou být zaměřeny jen na určitou zájmovou skupinu. Tedy například na stavebnictví, nebo jiné.
- **cena** - z pohledu živnostníků a malých firem jistě velice důležitý faktor. Pokud by služby byly příliš drahé, určitě by odradili spoustu uživatelů, kteří by se raději uchýlili ke konkurencím i za předpokladu, že by dané konkurenční služby nedosahovali takové kvality.
- **množství poptávek a nabídek** - pokud se uživatel zaregistruje jako dodavatel na poptávkový portál, vyžaduje, aby bylo na portálu dostatek poptávek, na které může reagovat. Z pohledu odběratele, jsou očekávány nabídky od dodavatelů.
- **škála nabízených funkcí** - jedná se o množství funkcí, které máme jako uživatel (registrovaný i neregistrovaný) k dispozici. Může se jednat o různé přehledy, funkce pro tisk a vše, co nám ulehčí a hlavně urychlí práci s portálem.
- **uživatelská podpora** - je důležité, aby portál poskytl uživateli dobrou a rychlou podporu v případě, kdy uživatel potřebuje poradit, nebo urychleně vyřídit důležité

záležitosti. Podpora může být technického rázu prostřednictvím e-mailů, či telefonních služeb. Uživatelé také ocení uživatelskou příručku, nebo alespoň sekci často kladených otázek.

- **uživatelské prostředí** - mnoho potencionálních zákazníků se rozhoduje dle prvního dojmu. I velice funkčně propracovaný portál může narazit, pokud nezaujme na první pohled svým grafickým zpracováním, uspořádáním prostředí, či funkcí. Takový portál se může přichozímu jevit jako nedůvěryhodný, či zastaralý.

Nyní budeme analyzovat konkrétní poptávkové portály, na které můžeme nejčastěji narazit na českém trhu.

3.3.1 Eoptavka.cz

Jedná se o jeden z nejstarších a nejznámějších poptávkových portálů v České Republice, dostupný na adrese <http://www.epoptavka.cz>. Podle údajů, které jsou uváděny na stránkách portálu, se může pyšnit i největší databází poptávek. Nabízeny jsou čtyři tarify pro dodavatele, registrace odběratelů není možná a vkládání poptávek je na portálu zdarma. Služba je u všech tarifů totožná. Jediným rozdílem je cena a doba platnosti, jak ukazuje tabulka 1.

Služba	Platnost služby	Cena
Tarif Na zkoušku	3 měsíce	1 290 Kč
Tarif Standard	6 měsíců	1 990 Kč
Tarif Profi	12 měsíců	2 990 Kč
Tarif Profi + Veřejné poptávky a zakázky	12 měsíců	4 990 Kč

Tabulka 1: Tarify, nabízené portálem Eoptavka.cz

Dodavatel registrovaný na tomto portálu má v placené verzi přístup k poptávkám z vybraného oboru a je mu garantováno zasílání nových poptávek z těchto kategorií. Po registraci obdrží dodavatel elektronickou fakturu na zadaný e-mail a během několika hodin je zasílání poptávek plně aktivní. Pokud dodavatel uhradí zaslanou fakturu, jsou mu prostřednictvím e-mailu zaslány přístupové kódy, díky kterým se může přihlašovat do zákaznické sekce. Zde může využívat archiv všech poptávek, či nastavovat profil. Služba také nabízí týdenní zasílání veřejných zakázek z celé Evropy [7].

Pokud se jako dodavatel zaregistrujete, portál epoptavka.cz vám nabídne následující funkce:

- okamžitý přístup k vybraným poptávkám dle objednaného tarifu
- zasílání odpovídajících a ověřených poptávek přímo na váš e-mail
- telefonickou i e-mailovou podporu služby
- možnost nastavení vlastního profilu pro zasílání poptávek

Obrázek 1: Epopstavka.cz – ukázka portálu

- bezplatnou registraci do systému B2M

Celkově je portál ePoptavka.cz graficky poměrně dobře zpracován, je přehledný a dobře ovladatelný. Na obrázku 1 je vidět úvodní stránka portálu.

3.3.2 AAA poptávka

Jedná se také o jeden z nejnavštěvovanějších poptávkový portálů v České Republice, dostupný na adrese <http://www.aaapoptavka.cz>. Týdně je na portál vloženo okolo 1500 nových poptávek. Ty je možno vložit zdarma. Tento portál neumožňuje registraci odběratelů. AAA poptávka nabízí pro dodavatele hned několik tarifů a balíčků služeb, které jsou rozlišeny podle toho, co přesně chcete na poptávkovém portálu využívat. Tarify jsou vypsány v tabulce 2, balíčky služeb pak v tabulce 3. Základní registrace je pro dodavatele zdarma, nebudete mít ovšem přístup ke kontaktům odběratelů. Portál také nabízí získávat poptávky ze slovenského serveru www.aaadopyt.sk a tím umožňuje dodavatelům rozšířit svou působnost i do zahraničí. Dodavatelé mohou být zároveň odběrateli. Mohou totiž vkládat neomezené množství poptávek a následně je spravovat v klientské sekci. Tím mohou nabízet a poptávat z jednoho uživatelského účtu.

Portál garantuje dodavatelům, že zveřejní 30000 poptávek ročně, pokud se tak nestane, vrátí uživateli poměrnou část registračního poplatků. Registrace je tedy pro dodavatele bez rizika, co se týká počtu poptávek [8].

Jak je vidět na obrázku 2, grafické zpracování portálu je poněkud jednoduché. Ovladatelnost a přehlednost je však celkem dobrá.

Po registraci na AAA poptávku tedy získáte následující výhody:

- zasílání poptávek na e-mail
- zasílání veřejných zakázek malého rozsahu a podlimitních veřejných zakázek na e-mail
- zápis firmy do katalogu firem
- vkládání a správa poptávek

Tarif	6 měsíců	12 měsíců	24 měsíců
Přístup k soukromým a firemním poptávkám	1990 Kč	2490 Kč	3990 Kč
Přístup na veřejné zakázky malého rozsahu	1990 Kč	2490 Kč	3990 Kč
Firemní zápis PROFI	1499 Kč	1999 Kč	2999 Kč
Přístup ke slovenským poptávkám	1499 Kč	1999 Kč	2999 Kč

Tabulka 2: Tarify, nabízené portálem AAA poptávka

Balíčky služeb	12 měsíců	24 měsíců
Poptávky a veřejné zakázky malého rozsahu	3999 Kč	5999 Kč
Poptávky a firemní zápis PROFI	3499 Kč	5499 Kč
Veřejné zakázky malého rozsahu a firemní zápis PROFI	3499 Kč	5499 Kč
Poptávky, veřejné zakázky malého rozsahu a firemní zápis PROFI	4999 Kč	7999 Kč

Tabulka 3: Balíčky služeb, nabízené portálem AAA poptávka

3.3.3 NiceOne

NiceOne je jeden z mála poptávkových portálů zaměřujících se pouze na určité odvětví. V tomto případě na stavebnictví. Můžete zde očekávat poptávky od podlahových prací až po dům na klíč. I když je tento web jeden z nejstarších a zároveň i nejdražších, je firmami a soukromými osobami hojně využíván. Grafické zpracování a funkce systému jsou opravdu strohé. Základními funkcemi jsou zasílání poptávek na e-mail, zápis do firemního katalogu, vkládání poptávek. Navíc portál nabízí možnost částečné registrace poptávajícího, která probíhá při vkládání poptávek a díky ní má uživatel přístup k editaci poptávky i později.

Tento portál přináší i nový systém u tarifů pro dodavatele. Tarify se totiž neodlišují pouze dle délky platnosti. Rozlišeny jsou také dle výše cenové relace poptávek, ke kterým budete mít přístup. Proto při objednání základního tarifu budete moci přistupovat

Obrázek 2: AAA poptávka – ukázka portálu

k poptávkám, které mají obrát od 0 do 40 000,- Kč. Vyšší tarif vám zaručí přístup k poptávkám do 100 000,- Kč a nejvyšší nad 100 000,- Kč. Každý tarif také garantuje různý počet poptávek. Od 30 kusů za rok u základního, až po 100 ks u nejvyššího tarifu. Ceny za tarify však přímo uváděny nejsou. Ty se vypočítávají v závislosti na počtu poptávek a firem pro daný region a obor, což je další specifikum tohoto portálu oproti konkurenci [9]. Veřejné poptávky tento portál neposkytuje. Portál je dostupný na adrese <http://www.niceone.cz>.

3.3.4 Ostatní

I ostatní poptávkové portály nabízí stejné, či velice podobné služby. Mezi ty významnější na českém trhu, můžeme zařadit ještě Poptávky.cz (<http://www.poptavky.cz>) a portál Poptávka.cz (<http://www.poptavka.cz>).

Zajímavostí je ještě portál GeM (dostupný na <http://www.poptavam.cz>). Ten se zaměřuje pouze na veřejné zakázky, které jsou určeny pro obchodování mezi subjekty veřejné správy a dodavateli. GeM nabízí organizaci elektronických výběrových řízení a nákupních aukcí na široký sortiment výrobků a služeb.

3.4 Specifikace nového systému

V kapitole 4 se budeme věnovat detailnímu specifikování požadavků. Před tím než k této počáteční fázi přistoupíme, nastíníme základní představu zadavatele na vyvíjený systém. Tato představa bude v následující fázi upravena a rozšířena.

Cílem je vytvořit poptávkový portál s administračním systémem pro správu systému. Portál bude všeobecně zaměřen, nebude tedy určen pouze pro jednu zájmovou skupinu.

Nový systém by měl obsahovat katalog firem, do kterého se budou moci firmy registrovat. Registrace bude obsahovat několik variant tarifů a tyto tarify bude možné měnit v administraci. Prezentace firmy bude nabízet funkce jako je umístění loga, galerií obrázků a příloh, přidávání poboček, atd. Katalog bude rozdělen do základních kategorií odvětví firem. Dané odvětví budou rozděleny v druhé úrovni na další pod odvětví. V katalogu bude umožněno také firmy vyhledávat. Systém bude obsahovat vkládání poptávek. Také poptávky budou děleny na kategorie (Auto-moto, IT software, atd.). Na úvodní stránce budou zobrazeny aktuální poptávky. Poptávky bude moci uživatel sortovat. U poptávek se bude zaznamenávat, kolikrát byla zobrazena. Firmy budou moci prohlížet poptávky a zasílat své nabídky.

Systém by měl umožňovat posílání osobních zpráv na danou poptávku, kterou uvidí pouze poptávající. Také budou moci uživatelé interně zasílat zprávy mezi sebou. Posílání e-mailů uživatelům s aktuálními nabídkami, poptávkami (vzhledem k oborům, které si uživatel zvolil při registraci), informace o registraci, atd.

Bude obsahovat také analytiky - grafy a souhrny některých statistik. Dále tiskové sestavy - možnost vytištění detailu nabídky, detailu poptávky, případně jiné. Export minimálně do formátu PDF.

Systém bude obsahovat administrační systém pro správu celého poptávkového portálu. Bude obsahovat částečný redakční systém pro možnost editace aktualit na portálu, kontaktů na portál, vkládání inzertních bannerů. Dále bude administrace zahrnovat správu poptávek, nabídek, kategorií, přehledy o registrovaných atd.

3.5 Inovace

Protože již na českém trhu několik poptávkových portálů funguje, je potřeba nabídnout produkt, který bude konkurence schopný a který nad ostatními vynikne. Proto je potřeba určit, jak poptávkový portál, který budeme vyvíjet, inovovat, odlišit.

Můžeme se například podívat na sociální sítě. Ty si v poslední době vydobýli obrovskou popularitu. Běžní uživatelé tráví v těchto online komunitách již tolik času, že se začíná hovořit o generační proměnně webu jako takového. Tyto sítě ale nejsou zaměřeny jenom na jednotlivce, ale také na firmy. Pro firmy je sociální síť místo pro přesně cílené reklamy, či jako účinný prostředek pro budování značky. Proto je velice žádoucí, aby se poptávkový portál jevil uživateli jako jeden z možných nástrojů komunikace. Z tohoto pohledu by měli dodavatelé-firmy díky poptávkovému portálu obsáhle prezentovat své aktivity a komunikovat s ostatními firmami, či odběrateli, přímo na samotném portálu. Ostatní poptávkové portály tuto komunikaci obcházejí pomocí zasílání e-mailů, což bohužel často vede k tomu, že se další komunikace mezi odběratelem a dodavatelem

odehrává mimo portál. Půjde tedy o to, vytvořit systém interních zpráv, jako je tomu u sociálních sítí. Tím by uživatelé na portálu trávili i více času a návštěvnost webu by vzrostla, což by se posléze dalo využít jako argument pro reklamní, ale i jiné účely.

Dnešním trendem je moderní a dynamický web. Je tedy potřeba vytvořit stránky pro uživatele atraktivní, které budou obsahovat interaktivní prvky, ale zároveň musí být pro uživatele přehledné. Proto bude dbáno na kvalitu vzhledu. Vhled stránek je totiž první věcí, které si uživatel všímá. Stránky, které zaujmou na první pohled, zaručují, že se s velkou pravděpodobností uživatel na web vrátí. Také rozmístění jednotlivých komponent, musí být pro uživatele intuitivní, aby se nemusel učit novým věcem, ale mohl využít vše hned, jak je zvyklý.

Další inovací oproti konkurenčním poptávkovým portálům, bude bezesporu využití AJAX a Silverlight prvků. Díky těmto prvkům bude portál nabízet dynamiku a plynulost celého webu.

Velkou oblibu zaznamenaly elektronické platební systémy, které se dnes využívají při platbách za zboží a služby na internetu. Usnadňují a urychlují platební styk mezi zákazníkem a dodavatelem a jsou proto pro obě strany výhodné. Protože jsou uživatelé zvyklí, za poskytnuté služby zaplatit různými způsoby, bude jim poskytnut i systém elektronických plateb. Při analýze konkurenčních poptávkových portálů bylo navíc zjištěno, že tuto možnost žádný ze zkoumaných portálů neposkytuje.

4 Specifikace požadavků

Specifikace požadavků má za úkol popsat co má softwarový systém dělat. Zaměřujeme se tedy na specifikaci jeho funkcionality. Výsledkem této fáze jsou modely specifikace požadavků, které slouží k odsouhlasení zadání mezi vývojovým týmem a zadavatelem [2].

Pokud si rozdělíme aktivitu spojenou s požadavky, budeme postupovat dle následující posloupnosti:

- identifikace funkčních požadavků
- specifikace případů užití
- specifikace nefunkčních požadavků

4.1 Identifikace funkčních požadavků

Jazyk UML neposkytuje žádné doporučení, které by se týkalo psaní specifikací systémových požadavků. S požadavky se vypořádává výhradně prostřednictvím mechanismu případů užití. Přesto se však mnoho analytiků a návrhářů domnívá, že případy užití samy nestačí a že potřebujeme specifikaci systémových požadavků [3].

Pro vyjádření požadavků proto využijeme jednoduchého formátu. Každý požadavek bude obsahovat jedinečný identifikátor, klíčové slovo a příkaz funkce. Ke každému požadavku, bude přiřazena i priorita. Tato hodnota značí relativní prioritu požadavku vůči ostatním požadavkům. Využijeme následující množiny kritérií zachycenou v tabulce 4. Je-li použito těchto kritérií, budeme atribut *Priorita* značit hodnotami: M, S, C, nebo W.

Hodnoty atributu priorita	Sémantika
Nezbytný (Must have)	Povinné požadavky, jež jsou základem systému.
Možný (Should have)	Důležité požadavky, které lze však vynechat.
Eventuální (Could have)	Požadavky, jež jsou opravdu nepovinné (je-li čas).
Chceme mít (Want to have)	Požadavky mohou být zahrnuty do dalších verzí systému.

Tabulka 4: Priority požadavků

Nejdříve však požadavky uspořádáme do skupin dle typu, což je užitečné obzvláště v případě, kdy pracujeme s větším množstvím požadavků. Těchto skupin můžeme následně využít k rozdělení systému na jednotlivé moduly. Rozdělení může vypadat následovně:

Funkční požadavky:

- Zákazníci
- Uživatelé
- Poptávky

- Nabídky
- Zprávy
- Pobočky
- Tarify
- CMS
- Analytiky
- Mailing
- Faktury - platby

Nyní budeme jednotlivé typy požadavků dále specifikovat. Význam většiny požadavků je již svým klíčovým slovem a příkazem funkce jasný. Budeme se tedy zaměřovat na popis nejasných požadavků, či na jiné důležité informace vztahující se k typům požadavků.

4.1.1 Funkční požadavky - Zákazníci

V tabulce 5 jsou specifikovány všechny požadavky na typ *Zákazníci*. Zákazníkem je každá registrovaná entita, tedy všichni dodavatelé a odběratelé. V tabulce lze vidět, že požadavky vztahující se ke galerii a fotografiím, jsou dle atributu priority nepovinné. Ostatní jsou povinné a jsou základem systému. Editace loga je důležitým požadavkem, ale lze jej vynechat.

4.1.2 Funkční požadavky - Uživatelé

Tento typ sdružuje všechny požadavky, které se týkají uživatelů systému. Uživatel je každá osoba, která má vytvořen účet a má tak přístup do zákaznické sekce. Vztah zákazníka a uživatele je tedy takový, že zákazník může vytvořit jeden a více uživatelských účtů. K jednomu zákaznickému účtu tak může přistupovat více uživatelů. Mezi povinné požadavky nesmí chybět funkce pro zapomenuté heslo a funkce pro správu logů. Požadavky jsou v tabulce 6.

4.1.3 Funkční požadavky - Poptávky

Funkční požadavky na poptávky jsou zobrazeny v tabulce 7. Poptávky budou klíčovou doménou celého systému. Mezi rozšiřující požadavky patří blokace poptávky či uzavření poptávky. Dále je požadována funkce pro nahlášení nevhodných poptávek uživateli. Je požadována také tisková sestava pro poptávku, či funkce pro správu kategorií.

4.1.4 Funkční požadavky - Nabídky

Na poptávky bude možno reagovat nabídkami. Proto nejdůležitějším požadavkem typu *Nabídka* je *Vložení nabídky*. Dále je požadována funkce pro vložení přílohy a požadavek na tiskovou sestavu nabídky. Požadavky se nachází v tabulce 8.

4.1.5 Funkční požadavky - Zprávy

V tabulce 9 se nachází požadavky typu *Zpráva*. Jedná se o požadavky na interní zasílání zpráv v systému. Zprávy by měli být zasílány mezi uživateli a také mezi systémem a uživatelem. Tyto požadavky mají vysokou prioritu.

4.1.6 Funkční požadavky - Pobočky

Pobočky rozšiřují nastavení dodavatelů. Díky tomu, mají firmy možnost prezentovat i své pobočky v rámci jednoho zákaznického účtu. Požadavky jsou v tabulce 10. Mezi ty nejdůležitější patří požadavky pro správu poboček. Dále je požadována správa e-mailů, telefonních čísel, webových adres a otevírací doby pobočky.

4.1.7 Funkční požadavky - Tarify

Pro kompletní správu systému je potřeba také upravovat nabízené tarify. Proto je požadována také kompletní správa tarifů. *Generování benefitu* je požadavek s nízkou prioritou. Ten by měl generovat klíče, které budou nabízeny novým potencionálním zákazníkům. Ti by tyto klíče využili při registraci a tím získali odpovídající slevu na tarif. Požadavky typu *Tarif* se nachází v tabulce 11.

4.1.8 Funkční požadavky - CMS

Pro správu obsahu bylo požadováno vkládání aktualit a bannerů. Tyto požadavky zadavatele byly rozšířeny tak, aby vznikl ucelený modul redakčního systému, který bude nabízet kompletní správu obsahu a bude možné jej v případě potřeby použít i v jiných projektech. Požadavky, které mají nižší prioritou, jsou Správa článku, Správa kategorií, Správa sekcí a Správa menu položek. Tento typ požadavků se nachází v tabulce 12.

4.1.9 Funkční požadavky - Analytiky

Pro lepší přehled toho, co se děje v systému, slouží analytiky. Mezi požadavky patří statistika poptávek, která by měla zobrazovat přehledy vložených poptávek, statistika dodavatelů a statistika odběratelů a statistika návštěvnosti. Tyto požadavky jsou zobrazeny v tabulce 13.

4.1.10 Funkční požadavky - Mailing

Jak můžeme vidět v tabulce 14, tento typ požadavků má nižší prioritu. Bude dále rozšiřován zadavatelem. Jde o požadavky na zasílání e-mailů klientům, hromadné zasílání e-mailů a výpisy odeslaných a neodeslaných e-mailů.

4.1.11 Funkční požadavky - Faktury - platby

Systém bude po objednání služby vystavovat Výzvu k platbě. Po zaplacení bude vystavena faktura. Zákazník bude moci výzvu i fakturu tisknout, či exportovat. Platební modul bude rozšiřován zadavatelem v dalších verzích systému. Pro ucelení identifikace požadavků jsou zde však požadavky na platby uvedeny. Všechny požadavky jsou zobrazeny v tabulce 15.

ID	Klíčové slovo	Funkce	Priorita
	Registrace dodavatele	Registruje dodavatele	M
	Registrace odběratele	Registruje odběratele	M
	Výpis zákazníků	Vypíše všechny zákazníky	M
	Výpis dodavatelů	Vypíše všechny dodavatele	M
	Výpis odběratelů	Vypíše všechny odběratele	M
	Dodavatelé v kategorii	Zobrazí dodavatele v dané kategorii	M
	Detail zákazníka	Zobrazí detail zák.	M
	Editace zákazníka	Edituje údaje zákazníka	M
	Smazání zákazníka	Vymaže zákazníka	M
	Změna verifikace	Změní verifikaci	M
	Zobrazení katalogu	Zobrazí kategorie dodavatelů	M
	Nová kategorie	Vytvoří kategorii v katalogu	M
	Editace kategorie	Edituje kategorii v katalogu	M
	Smazání kategorie	Smaže kategorii v katalogu	M
	Editace loga	Edituje logo	S
	Vložení galerie	Vytvoří novou galerii	C
	Editace galerie	Edituje galerii	C
	Smazání galerie	Smaže galerii	C
	Vložení fotografie	Vloží fotografii	C
	Editace fotografie	Úpraví fotografii	C
	Smazání fotografie	Smaže fotografii	C

Tabulka 5: Funkční požadavky - Zákazníci

4.2 Specifikace případů užití

V jazyce UML se u specifikace požadavků využívá Use Case diagramu, neboli diagramu případu užití, který patří mezi modely, které popisují funkcionalitu systému. Tento diagram slouží ke vhodnému nastínění, jakou bude mít systém ve výsledku funkcionalitu.

ID	Klíčové slovo	Funkce	Priorita
	Výpis všech uživatelů	Vypíše všechny uživatele	M
	Výpis kontaktních osob	Vypíše všechny kontaktní osoby	M
	Výpis uživatelů pobočky	Vypíše uživatele pod danou pobočku	M
	Výpis administrátorů	Vypíše administrátory systému	M
	Vložení uživatele	Přidá nového uživatele	M
	Přiřazení k pobočce	Přiřadí uživatele k dané pobočce	M
	Editace uživatele	Upraví údaje uživatele	M
	Detail uživatele	Zobrazí detail uživatele	M
	Smazání uživatele	Smaže uživatele	M
	Změna hesla	Změní heslo	M
	Zapomenuté heslo	Vytvoří nové heslo a zašle jej uživateli	M
	Výpis uživatelů zákazníka	Zobrazí všechny uživatele zákazníka	M
	Log uživatele	Přidá log uživatele	M
	Smazání logu	Smaže log	M
	Výpis logů uživatele	Vypíše všechny logy	M

Tabulka 6: Funkční požadavky - Uživatelé

ID	Klíčové slovo	Funkce	Priorita
	Detail poptávky	Zobrazí detail poptávky	M
	Poptávky zákazníka	Vypíše poptávky zákazníka	M
	Blokace poptávky	Zablokuje poptávku	M
	Deblokace poptávky	Odblokuje poptávku	M
	Uzavření poptávky	Uzavře poptávku	M
	Počet nabídek	Zjistí počet nabídek na poptávku	M
	Nahlášení poptávky	Zapíše podezřelou poptávku	C
	Výpis hlášených poptávek	Vypíše všechny hlášené poptávky	C
	Smazání hlášené poptávky	Vymaže hlášenou poptávku	C
	Výpis kategorií poptávek	Zobrazí všechny kategorie poptávek	M
	Výpis poptávek v kategorii	Zobrazí poptávky v dané kategorii	M
	Tisková sestava poptávky	Zobrazí poptávku pro tisk	S
	Nová kategorie poptávek	Vytvoří kategorii poptávek	S
	Vložení ikony	Vloží ikonu ke kategorii	C
	Editace kategorie poptávky	Edituje kategorii poptávky	S
	Smazání kategorie poptávek	Smaže kategorii poptávek	S

Tabulka 7: Funkční požadavky - Poptávky

ID	Klíčové slovo	Funkce	Priorita
	Nabídky zákazníka	Vypíše nabídky zákazníka	M
	Detail nabídky	Zobrazí detail nabídky	M
	Výpis nabídek k poptávce	Vypíše všechny nabídky k poptávce	M
	Vložení přílohy	Vloží přílohy	S
	Tisková sestava nabídky	Zobrazí nabídku pro tisk	S
	Vložení nabídky	Vloží novou nabídku	M

Tabulka 8: Funkční požadavky - Nabídky

ID	Klíčové slovo	Funkce	Priorita
	Poslat zprávu	Zašle zprávu vybranému uživateli	M
	Poslat zprávu o registraci	Zašle zprávu o registraci	M
	Poslat zprávu o nové poptávce	Zašle zprávu o nové poptávce	M
	Poslat zprávu o nové nabídce	Zašle zprávu o nové nabídce	M
	Odpovědět na zprávu	Odešle odpověď na zprávu	S
	Detail komunikace	Zobrazí detail komunikace	S

Tabulka 9: Funkční požadavky - Zprávy

ID	Klíčové slovo	Funkce	Priorita
	Výpis poboček dodavatele	Vypíše všechny pobočky dodavatele	M
	Detail pobočky	Zobrazí detail pobočky	M
	Editace pobočky	Upraví pobočku	M
	Nová pobočka	Vytvoří novou pobočku	M
	Změna viditelnosti	Změní viditelnost pobočky	S
	Nový e-mail	Vytvoří nový e-mail	S
	Editace e-mailu	Upraví e-mailu	S
	Smazání e-mailu	Smaže e-mail	S
	Nové telefonní číslo	Vytvoří nové telefonní číslo	S
	Editace telefonního čísla	Upraví telefonní číslo	S
	Smazání telefonního čísla	Smaže telefonní číslo	S
	Přidat www adresu	Přidá www adresu	S
	Editace www adresy	Upraví www adresu	S
	Smazání www adresy	Smaže www adresu	S
	Editace otevírací doby	Upraví otevírací doby	S

Tabulka 10: Funkční požadavky - Pobočky

ID	Klíčové slovo	Funkce	Priorita
	Nový tarif	Vytvoří nový tarif	S
	Editace tarifu	Upraví tarif	S
	Smazání tarifu	Smaže tarif	S
	Generování benefitu	Vygeneruje benefity	C
	Smazání benefitu	Smaže benefit	C
	Detail tarifu	Zobrazí detail tarifu	S

Tabulka 11: Funkční požadavky - Tarify

ID	Klíčové slovo	Funkce	Priorita
	Vložení aktuality	Vloží novou aktualitu	S
	Editace aktuality	Upraví aktualitu	S
	Smazání aktuality	Smaže aktualitu	S
	Publikovat aktualitu	Nastaví aktualitu pro publikování	C
	Vložení banneru	Vloží banner	S
	Editace banneru	Upraví banner	S
	Smazání banneru	Smaže banner	S
	Správa článků	Spravuje články	C
	Správa kategorií článků	Spravuje kategorie článků	C
	Správa sekcí článků	Spravuje sekce článků	C
	Správa menu položek	Spravuje menu položky	C

Tabulka 12: Funkční požadavky - CMS

ID	Klíčové slovo	Funkce	Priorita
	Statistika poptávek	Zobrazí statistiku poptávek	S
	Statistika dodavatelů	Zobrazí statistiku dodavatelů	S
	Statistika odběratelů	Zobrazí statistiku odběratelů	S
	Statistiky návštěvnosti	Různé statistiky návštěvnosti	S

Tabulka 13: Funkční požadavky - Analytiky

ID	Klíčové slovo	Funkce	Priorita
	Zaslání e-mailu	Zašle e-mail uživateli	C
	Zaslání hromadného e-mailu	Zašle hromadný e-mail	W
	Výpis odeslaných e-mailů	Zobrazí výpis odeslaných e-mailů	W
	Výpis neodeslaných e-mailů	Výpis neodeslaných e-mailů	W

Tabulka 14: Funkční požadavky - Mailing

Slouží také k základnímu vymezení hranic mezi systémem a okolím, tedy definuje, co existuje vně vyvíjeného systému (aktéři) a co má být systémem prováděno (případy užití). Notace tohoto modelu obsahuje tyto prvky:

- aktér (actor) - definující uživatele či jiné systémy, kteří budou vstupovat do interakce s vyvíjeným softwarovým systémem [2]
- hranice systému (system boundary) - vymezení hranice systému
- případ užití (use case) - specifikující vzory chování realizovaných softwarovým systémem. Každý případ užití lze chápat jako posloupnost vzájemně navazujících transakcí vykonaných v dialogu mezi aktérem a vlastním softwarovým systémem
- komunikace - vazba mezi aktérem a případem užití (aktér komunikuje se systémem na daném případě)

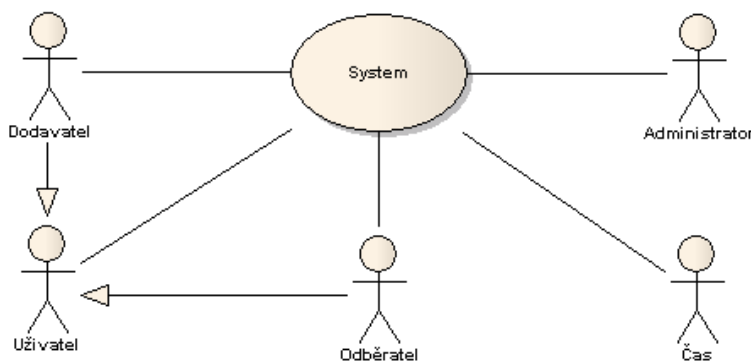
Pokud existuje část případů užití, které jsou společné více případům, můžeme jej v UML vyjmout do samostatného případu užití a ostatní případy užití se na něj budou moci odkazovat pomocí relace «include». V případě, že budeme chtít rozšířit chování nějakého případu užití jiným případem, můžeme použít «exclude».

4.2.1 Označení aktérů

Před tím než označíme aktéry, je dobré zvážit, kdo nebo co daný systém používá a jakou roli při komunikaci se systémem hraje. Existuje několik pomocných otázek, které pomáhají při určení aktérů. Například kdo nebo co používá daný systém? Jakou roli v této interakci hraje? Děje se něco v určitou dobu? Poté již můžeme vyhledat všechny aktéry v systému (viz. obrázek 3).

ID	Klíčové slovo	Funkce	Priorita
	Vystavení výzvy k platbě	Vystaví výzvu k platbě	C
	Vystavení faktury	Vystaví fakturu	C
	Tisková sestava výzvy k platbě	Zobrazí výzvu k platbě pro tisk	C
	Tisková sestava faktury	Zobrazí fakturu pro tisk	C
	Výpis výzev zákazníka	Zobrazí výzvy k platbě zákazníka	C
	Výpis faktur zákazníka	Zobrazí faktury zákazníka	C
	Výpis plateb zákazníka	Zobrazí platby zákazníka	C
	Výpis faktur	Zobrazí všechny faktury v systému	C
	Výpis výzev k platbě	Zobrazí všechny výzvy k platbě v systému	C
	Platební systém	Obecný požadavek na elektronický platební systém	C

Tabulka 15: Funkční požadavky - Faktury - platby



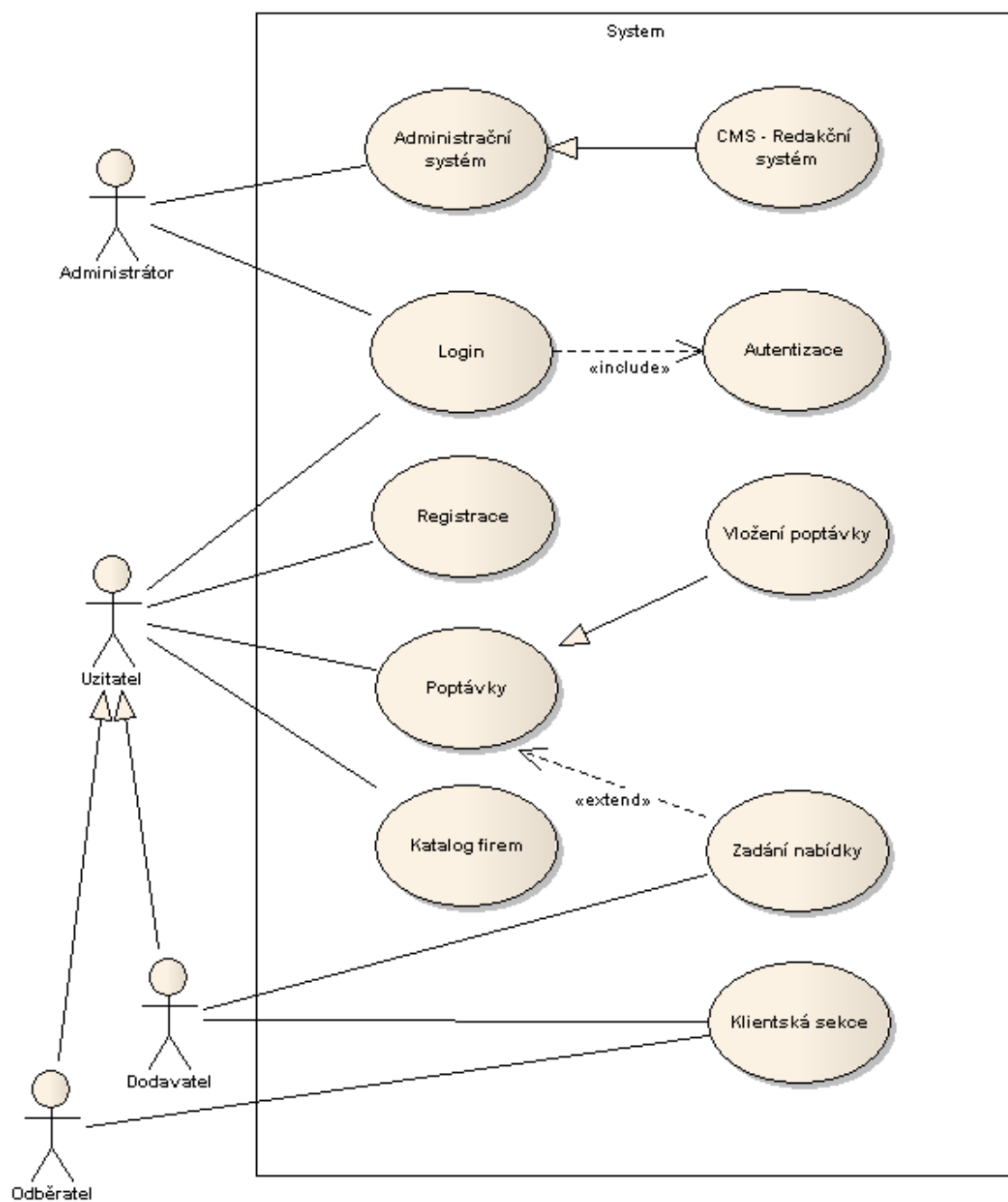
Obrázek 3: Use Case diagram – Kontext

Pro náš systém byli zjištěni následující čtyři aktéři:

- **Uživatel** - jedná se o běžného uživatele, který prochází systémem.
- **Odběratel** - aktér je rozšířením Uživatele. Je již registrován do systému a má přístup do zákaznické sekce.
- **Dodavatel** - také je rozšířením aktéra Uživatele. Jedná se o registrovanou firmu, zapsanou v katalogu firem, která má přístup do zákaznické sekce.
- **Administrátor** - aktér s nejvyššími právy. Spravuje celý systém.
- **Čas** - aktér, který spouští akce v určitém časovém bodě.

4.2.2 Systém

Na obrázku 4 je Use Case diagram celého systému. Systém je rozdělen na dvě části. První částí je administrační rozhraní, ke které mají přístup pouze administrátoři a mohou zde spravovat celý systém. Administrátoři mohou využívat i redakčního systému, díky kterému mohou upravovat obsahovou stránku webu, tzn. měnit menu, přidávat aktuality, či měnit kontaktní údaje. Druhou částí je samotný poptávkový portál, ke kterému mají přístup všichni uživatelé. Ti mohou procházet poptávky a katalogy firem, či mohou vkládat bez registrace a zdarma své vlastní poptávky. Do systému se mohou uživatelé také zaregistrovat. Registrovat se mohou uživatelé buď jako dodavatelé, což je placená služba, nebo jako odběratelé zdarma. Po registraci mají odběratelé i dodavatelé přístup do klientské sekce, která nabízí další administrativní možnosti.



Obrázek 4: Use Case diagram – Systém

4.2.3 Administrační systém

Administrační systém umožňuje administrátorům spravovat celý systém. Na obrázku 5 vidíme Use Case diagram tohoto případu.

Správa zákazníků sdružuje všechny registrované dodavatele a odběratele. Je možné je selektovat, editovat základní údaje a mazat. Lze také měnit verifikaci zákazníka, díky čemuž můžeme manuálně povolit, či zakázat přístup zákazníkovi. U dodavatelů je umožněna správa informací, poboček, uživatelských účtů, které spadají pod daného dodavatele, správa nabídek a poptávek dodavatele. U poptávek je samozřejmostí zobrazení nabídek, které byly přidány k dané poptávce. Správa poboček zahrnuje editaci údajů o pobočce, správu e-mailů, telefonních čísel, www stránek a otevírací doby pobočky. U odběratele lze spravovat základní informace, spravovat uživatelský účet a také poptávky stejně jako u dodavatele. Správa kategorií umožňuje vkládat, měnit a mazat kategorie v katalogu firem a umožňuje sledovat, kolik dodavatelů je pod kategoriemi zapsáno. Tyto kategorie jsou hierarchicky děleny na hlavní kategorie a podkategorie.

Správa uživatelů představuje souhrn všech uživatelských účtů, které jsou využívány dodavateli, či odběrateli, a díky kterým se uživatelé přihlašují do zákaznické sekce. Při registraci zákazníka se nastaví pouze jeden uživatelský účet. Dodavatelé, či odběratelé si tak následně mohou v zákaznické sekci vytvořit více uživatelských účtů. Do zákaznické sekce může mít poté přístup více pověřených osob zákazníka. Můžeme také u účtů měnit práva.

Správa poptávek umožňuje selektovat, editovat a mazat poptávky. Administrátor může také poptávku blokovat, či poptávku uzavřít. Pokud poptávka obsahuje nějaké nabídky, lze zobrazit jejich detail. Hlášené poptávky představují upozornění uživatelů na nevhodné, či nekorektní poptávky. Administrátor proto může po uvážení poptávku zablokovat. Hlášené poptávky lze také mazat.

Správa kategorií poptávek umožňuje vkládat, měnit a mazat kategorie poptávek. Tyto kategorie jsou pouze jednoúrovňové, což je rozdíl oproti kategoriím katalogu firem. Lze k nim však vkládat ikony.

Tarify portálu nejsou nastaveny napevno. Proto díky správě tarifů může administrátor s tarify manipulovat. Administrátor může vkládat nové tarify, upravovat již stávající, případně je mazat. Tarifům můžeme kromě ceny standardní nastavit i cenu při slevě. Systém totiž nabízí možnost generování slev - zvýhodněných kódů. Pokud je kód zadán při registraci, uplatní se při fakturaci sleva, která je u tarifu nastavena. Lze také sledovat, zda je již kód využit, či generovat nové nebo je mazat.

Správa logů ukazuje pohyb uživatelů na poptávkovém portálu. Jsou zaznamenávány základní informace, jako jsou přihlášení, odhlášení a jiné.

Správa interních zpráv umožňuje zasílat interní zprávy a sledovat také historii interních zpráv. Díky tomu lze ze systému zasílat zprávy uživatelům a to jak jednotlivě, tak i hromadně. Mailing zastřešuje případy užití, které jsou spjaté se zasíláním e-mailů. Administrátoři mohou také sledovat statistiky systému. Dále může administrátor pracovat s fakturami a platbami. Faktury může vystavovat a tisknout. Platby lze sledovat a případně zadat platbu novou.



Obrázek 5: Use Case diagram – Administrační systém

4.2.4 CMS - Redakční systém

Diagram na obrázku 6 ukazuje Use Case diagram CMS systému. Tento redakční systém umožňuje spravovat obsah portálu a je součástí administračního rozhraní, ke kterému má přístup aktér administrátor. Jak můžeme vidět na obrázku, redakční systém obsahuje funkce pro správu sekcí, kategorií, článků, aktualit, bannerů a jednotlivých menu.

Články jsou do systému vkládány, mohou být upravovány a mazány. Při vkládání článků je potřeba ještě vybrat sekci a kategorii, ke které bude článek patřit. Je také možné článek označit jako nekategorizovaný.

Kategorie a sekce mohou být stejně jako články vkládány, modifikovány a mazány. U kategorie ještě volíme, ke které sekci kategorie patří. Pomocí sekcí a kategorií nám vzniká jakási vnitřní hierarchie článků.

Každé menu obsahuje funkce pro správu položek menu. Přidání, editace a mazání jsou samozřejmostí. Navíc můžeme ještě měnit pořadí položek v menu. U položky menu se také volí typ položky. Změnou tohoto typu, můžeme k menu položce přiřadit sekci, kategorii, nebo pouze samostatný článek. Díky tomu může mít každá položka v menu jiný typ obsahu. Také je možné přiřadit položce menu typ externího odkazu, či interního odkazu.

Systém také umožňuje správu aktualit, díky kterým se můžou rychle vkládat aktuální informace na portál a správu bannerů, která umožňuje vložení bannerů na místa určená pro reklamu.

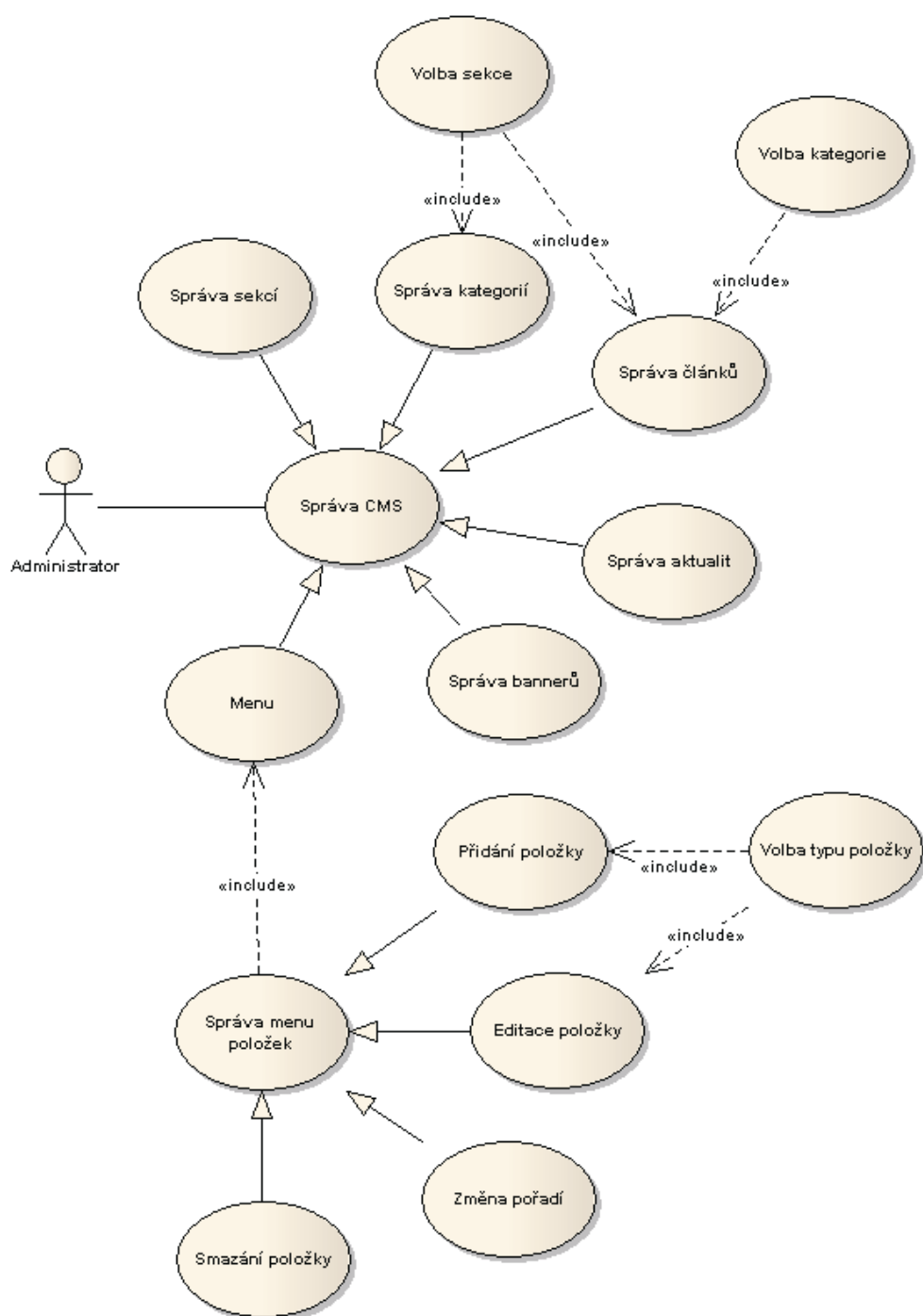
4.2.5 Zákaznická sekce

Do zákaznické sekce mají přístup všichni registrovaní zákazníci, tedy dodavatelé i odběratelé. Díky zákaznické sekci mohou dodavatelé měnit své údaje i hesla. Mohou zde také umístit logo firmy. Další funkcí je možnost vytváření galerií, do kterých se mohou vkládat prezentační fotografie. Jako dodavatel máme také možnost vrátet se k nabídkám, které jsme dříve sami vložili. Je také možno procházet vlastní poptávky a sledovat nabídky, které byly vloženy jako reakce od dodavatelů na naše poptávky. Dále můžeme zjišťovat detail firmy, která nabídku vložila a selektovat nabídky dle data vložení, či nabízené ceny. Zákaznická sekce dále umožňuje spravovat pobočky firmy, kde můžou uživatelé editovat e-mail, telefonní čísla, www stránky a otevírací doby pobočky. Uživatel je upozorňován pomocí zpráv o různých událostech, jako je nová nabídka a jiné, a tyto zprávy může i později sledovat, či mazat. Je zde také přehled faktur a výzev k platbám.

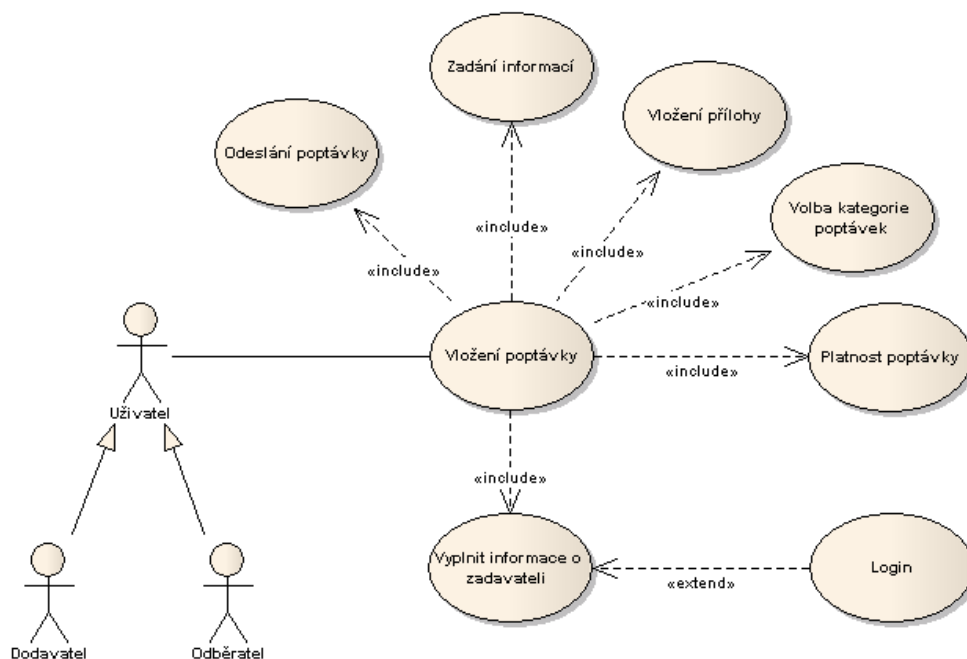
Odběratelé mají v zákaznické sekci omezenější možnosti. Mohou spravovat osobní údaje, spravovat své poptávky stejně jako dodavatelé a sledovat zprávy, které jim byli zaslány.

4.2.6 Zadání poptávky

Use Case diagram zadání poptávky je zobrazen na obrázku 7. Vkládat poptávku mohou neregistrovaní uživatelé a také registrovaní dodavatelé i odběratelé, kteří jsou zobecněním aktéra *Uživatel*. Při zadání poptávky musíme vyplnit informace týkající se poptávky, zvolit



Obrázek 6: Use Case diagram – CMS



Obrázek 7: Use Case diagram – Zadání poptávky

kategorii poptávek, do které bude námi vkládaná patřit, můžeme vložit přílohu a musíme zvolit platnost poptávky. Po vypršení platnosti se poptávka uzavře. Samozřejmostí je vyplnění údajů o zadavateli poptávky. Máme také možnost se přihlásit. Osobní údaje se tak vyplní automaticky.

Případ užití můžeme zachytit i v textové podobě. Při specifikaci případů užití bylo využito šablony, která obsahuje následující informace: název případu užití, jedinečný identifikátor případu užití (ID), stručný popis, aktéři zapojení do případu užití, vstupní podmínky, které musí být splněny ještě před spuštěním daného případu užití, hlavní scénář, alternativní scénář a výstupní podmínky, které musí být splněny na konci případu užití [3].

Zde je ukázka případu užití **Zadat poptávku**:

ID: 3.1

Stručný popis: Uživatel vkládá poptávku.

Primární aktéři: Uživatel (Dodavatel, Odběratel).

Vedlejší aktéři: Žádní.

Výstupní podmínky: Žádné.

Výstupní podmínky: Nová poptávka byla uložena do systému.

Hlavní scénář:

1. Příklad užití začíná, až Uživatel vybere „zadat poptávku“
2. Systém zobrazí formulář pro zadání poptávky
3. Dokud uživatel neodešle poptávku, systém validuje všechna formulářová pole
4. Uživatel vyplní informace o poptávce
5. Uživatel vloží přílohu k poptávce
6. Uživatel vybere kategorii, ke které poptávka patří
7. Uživatel vybere platnost poptávky
8. Uživatel vybere, zda je fyzická osoba, nebo firma
 - (a) KDYŽ uživatel vybere „fyzická osoba“
 - i. Systém zobrazí formulář pro fyzickou osobu
 - (b) KDYŽ uživatel vybere „firma“
 - i. Systém zobrazí formulář pro firmu
9. Uživatel vyplní informace o své osobě, či firmě
10. Uživatel odešle poptávku
11. Systém zkontroluje a uloží všechny vložené údaje do databáze
12. Systém zobrazí hlášku o úspěšném vložení

Alternativní scénář:

- 3a - 9a Uživatel může kdykoliv přerušit vkládání poptávky, stiskem tlačítka „zpět“
- 3a1 - 9a1 Systém ukončí provádění a přesměruje uživatele na titulní stránku
- 8a - 9a KDYŽ je uživatel přihlášen
- 8a1 - 9a1 Systém nebude zobrazovat formulář pro firmu, či fyzickou osobu
- 12a Systém zjistí závadu při vložení údajů do databáze
- 12a1 Systém zobrazí hlášku o neúspěchu a proces ukončí

4.3 Nefunkční požadavky

Nyní, když jsme specifikovali funkční požadavky pomocí diagramů případu užití (Use Case), je potřeba specifikovat i nefunkční požadavky, které specifikují vlastnosti nebo omezující podmínky v systému. Od zadavatele byly zjištěny následující požadavky:

- Celý systém bude napsán v technologii ASP.NET (C#) ve vývojovém prostředí Visual Studio 2008
- Bude použita databáze MS SQL Server 2008
- Při vytváření projektu bude využito technologií AJAX a Silverlight 3.0
- Architektura systému bude vícevrstvá
- Využití technologie LINQ
- Využití komponentových technologií
- Ohled na SEO optimalizaci - Rewrite Url pro tzv. hezké odkazy. Validita kódu
- Licenční politika s důrazem na zákonný průběh procesů systému

5 Analýza a návrh

Cílem této části bude pomocí analýzy a návrhu softwarového díla ukázat, jakým způsobem bude produkt realizován v implementační fázi [2]. Využijeme k tomu několika UML diagramů, které naznačí, jakým způsobem byla analýza a návrh prováděna.

5.1 Analýza

Záměrem analýzy je vytvořit analytický model. Tento model se zaměřuje na to, co systém musí udělat, avšak nezabývá se detaily týkajícími se způsobem, jakým to udělá [3].

Je tedy nutné zachytit problém z určité perspektivy a zabývat se abstrakcí problémové domény. Měla by však být zachována jednoduchost diagramů. V analýze se zaměříme spíše na vytváření logického modelu, kdy se budeme snažit zachytit nejdůležitější funkce, které musí systém obsahovat s cílem uspokojit zákazníka. V návrhu pak budou diagramy upraveny, aby bylo zřejmé, jakým způsobem tyto funkce budeme implementovat.

UML nabízí několik diagramů pro fázi analýzy a návrhu. Jako první využijeme třídni diagram, díky kterému můžeme ukázat nejdůležitější třídy, jejich vzájemné relace, klíčové atributy a operace, se kterými daná třída pracuje. Protože jsme ve fázi analýzy, budeme hledat analytické třídy, které modelují důležité aspekty problémové domény.

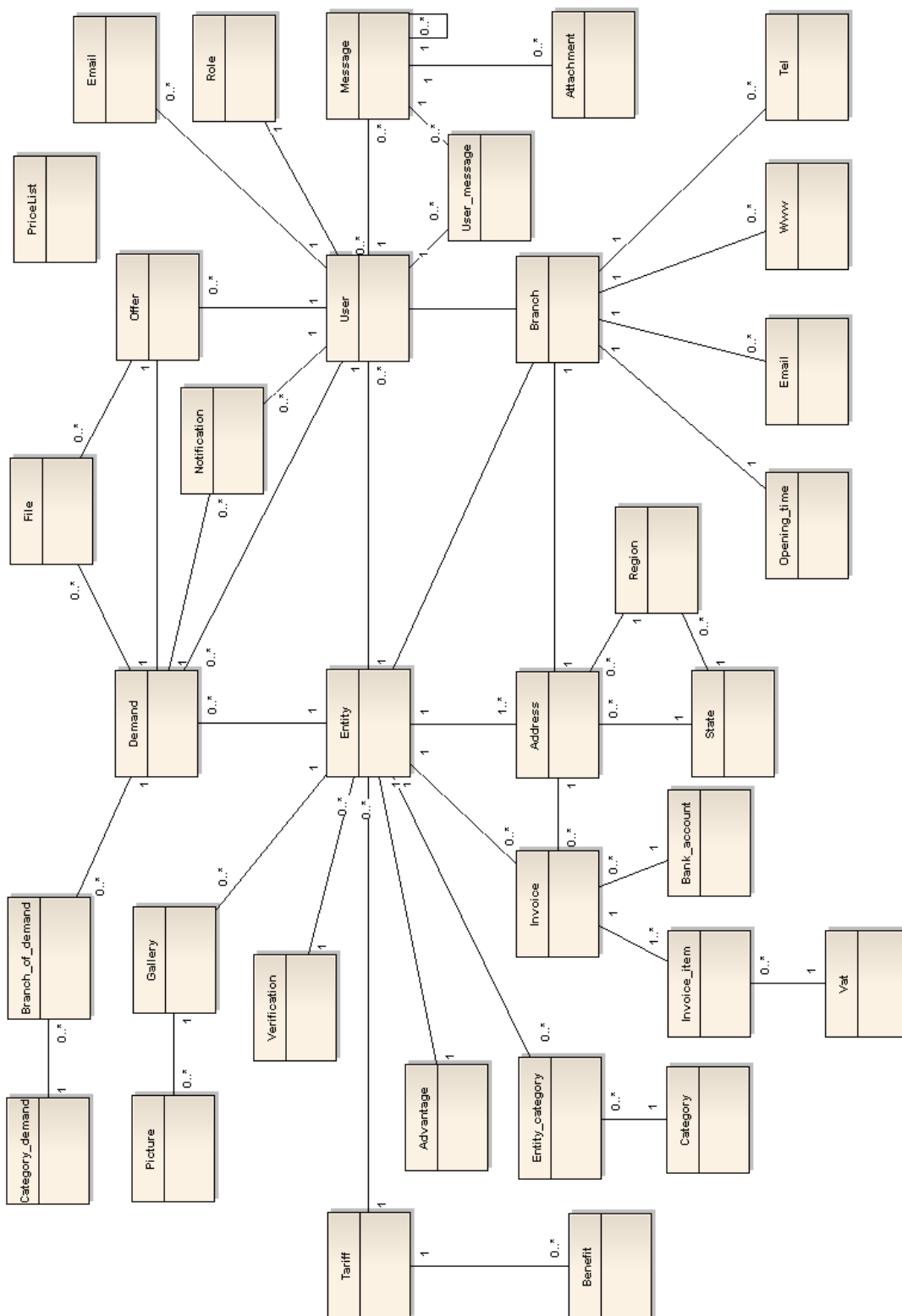
Na obrázku 8 vidíme již vytvořený třídni diagram, znázorňující systém. Jádrem systému je analytická třída *Customer* představující všechny zákazníky, tedy firmy a fyzické osoby. Dalšími klíčovými třídami je třída *User*, určující uživatelský účet, třída *Demand*, představující poptávku a *Offer*, což je nabídka.

5.2 Návrh

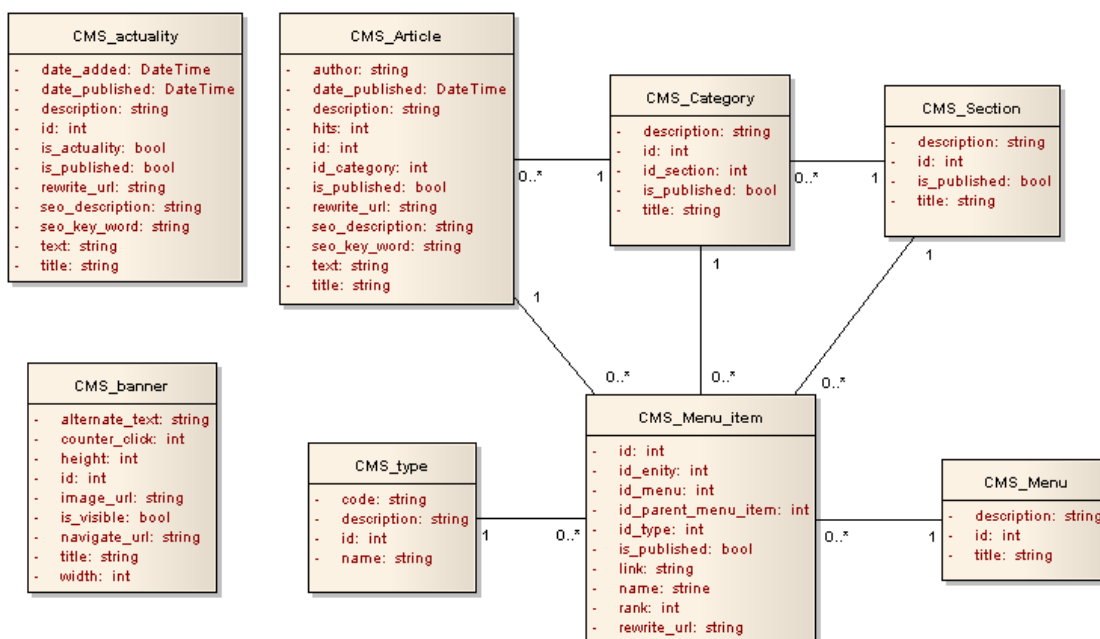
Ve fázi návrhu budeme model analýzy upravovat pro implementační prostředí. Protože se zabýváme webovou aplikací, budeme se snažit vytvořit datový model pro tvorbu relační databáze. Využijeme třídni diagramů, kdy třída bude představovat tabulku a asociace mezi třídami ekvivalentně vazby mezi tabulkami. Při následném využití nástroje LINQ to SQL, který mapuje tabulky v databázi na standardní .NET třídy, budeme mít totiž díky třídni diagramům zmapování celou strukturu, se kterou budeme pracovat při tvorbě databáze a po sléze i při samotné implementaci.

Na obrázku 9 je třídni diagram celého systému. Z diagramu je patrné rozložení jednotlivých tříd v systému a také vzájemné vazby mezi nimi. Protože je diagram příliš velký, jsou vypsány pouze názvy tříd. Jak již bylo zmiňováno, třídni diagram se shoduje s databázovou strukturou, proto jsou atributy obsaženy v datovém slovníku. Protože bude systém komerčně využíván, nebude v této práci podrobný datový slovník z bezpečnostních důvodů uveden. Na třídni diagramu je vidět význam třídy *Entity* pro celý systém. Tato třída sdružuje všechny dodavatele a odběratele v systému.

Součástí práce je také CMS systém. Na diagramu 10 je zobrazen třídni diagram reprezentující tuto část databáze. Na diagramu jsou označeny i atributy, které představují sloupce v databázi a také omezení kladené na jednotlivé tabulky. Popis atributů je uveden v datovém slovníku, který je součástí přílohy B.



Obrázek 9: Návrh databáze



Obrázek 10: Třídní diagram – CMS

5.2.1 Interakce a dynamické chování

Podstata specifikace dynamického chování objektově orientovaného systému spočívá v modelování komunikace mezi objekty. K tomu UML využívá několika diagramů. My využijeme sekvenční diagram a stavový diagram.

Sekvenční diagram zdůrazňuje časově orientovanou posloupnost zpráv mezi objekty. Tento diagram je přehledný, proto je často využíván díky lepšímu a rychlejšímu porozumění.

Stavový diagram je používán k popisu chování systému. Popisuje všechny možné stavy objektu a události, které způsobují přechod mezi jednotlivými stavy.

5.2.1.1 Odeslání poptávky Sekvenční diagramy postupují podle časové osy shora dolů, zatímco čáry života postupují zleva doprava. Čáry života jsou umístěny vodorovně, abychom minimalizovali počet překřížených čar v diagramu. Jejich svislé umístění vypovídá o okamžiku jejich vzniku. Pod každou čarou života je čerchovaná čára, která vyjadřuje délku čáry života v čase.

Sekvenční diagram na obrázku 11 znázorňuje způsob realizace chování případu užití *Odeslání poptávky*. V tomto případě je *Uživatel* objekt, který spouští komunikaci mezi objekty voláním synchronní zprávy *sendDemand*. Vytvoří se pomocný objekt *login*, který je instancí třídy *LoginInfo*. Následuje požadavek *checkLogin* na tento objekt, který zkontroluje, zda je uživatel přihlášen. Vrací se výsledek *true*. Jako reakce na tento výsledek pokračuje interakce požadavkem *getUserId*, který vrací identifikační číslo uživatele. S tímto ID je vo-

lána zpráva *findEntityByUser* na instanci třídy *Entities* představující databázovou tabulku a vrací objekt typu *Entity*. Poté je vyslán požadavek *new* na objekt *Demand_controller*, který zpracuje údaje. Tento objekt vytvoří pomocnou instanci *demand* třídy *Demand* a naplní ji daty. Dále vytváří a plní daty pomocné instance třídy *BranchOfDemand*. Následně jsou volány zprávy *insert* instancí tříd *Demands* a *Branch_of_demands*, které představují taktéž databázové tabulky. Po úspěšném uložení objekt třídy *Demand_controller* vrací výsledek akce instanci *DemandAdd*. Ten předává výsledek *Uživateli*.

5.2.1.2 Verifikace zákazníka Objekt *Zákazník* může v systému nabývat několika stavů. Na obrázku 12 je stavový diagram znázorňující životní cyklus tohoto objektu.

Po úspěšné registraci zákazník obdrží na e-mail verifikační kód, který musí být pro plnou aktivaci účtu zadán do systému. Jak vyplývá z diagramu, přechází proto objekt z počátečního stavu do stavu *Neverifikovaný* po obdržení zprávy o dokončené registraci. Pokud zákazník zadá aktivační kód, zasílá se systému zpráva *Zadej*, která způsobí přechod do stavu *Ověřování*. Tento stav je složený. Dal by se tedy vyjádřit více stavy. My však budeme předpokládat, že v tomto složeném stavu systém ověří pravost aktivačního kódu. Pokud je ověření úspěšné přejde objekt *Zákazník* do stavu *Verifikovaný*, pokud je ověření neúspěšné, objekt se vrací do stavu *neverifikovaný*. Pokud je zákazník verifikován, může jej administrátor zablokovat, například proto, že nebyly zaplacený objednané služby. Zasláním zprávy *Blokovat*, přechází objekt do stavu *Pozastavený*. Zavoláním zprávy *Odblokovat* se objekt vrací do stavu *Verifikovaný*. Zákazník může být ze systému i smazán. Proto po zavolání zprávy *Smazat*, přechází objekt do koncového stavu.

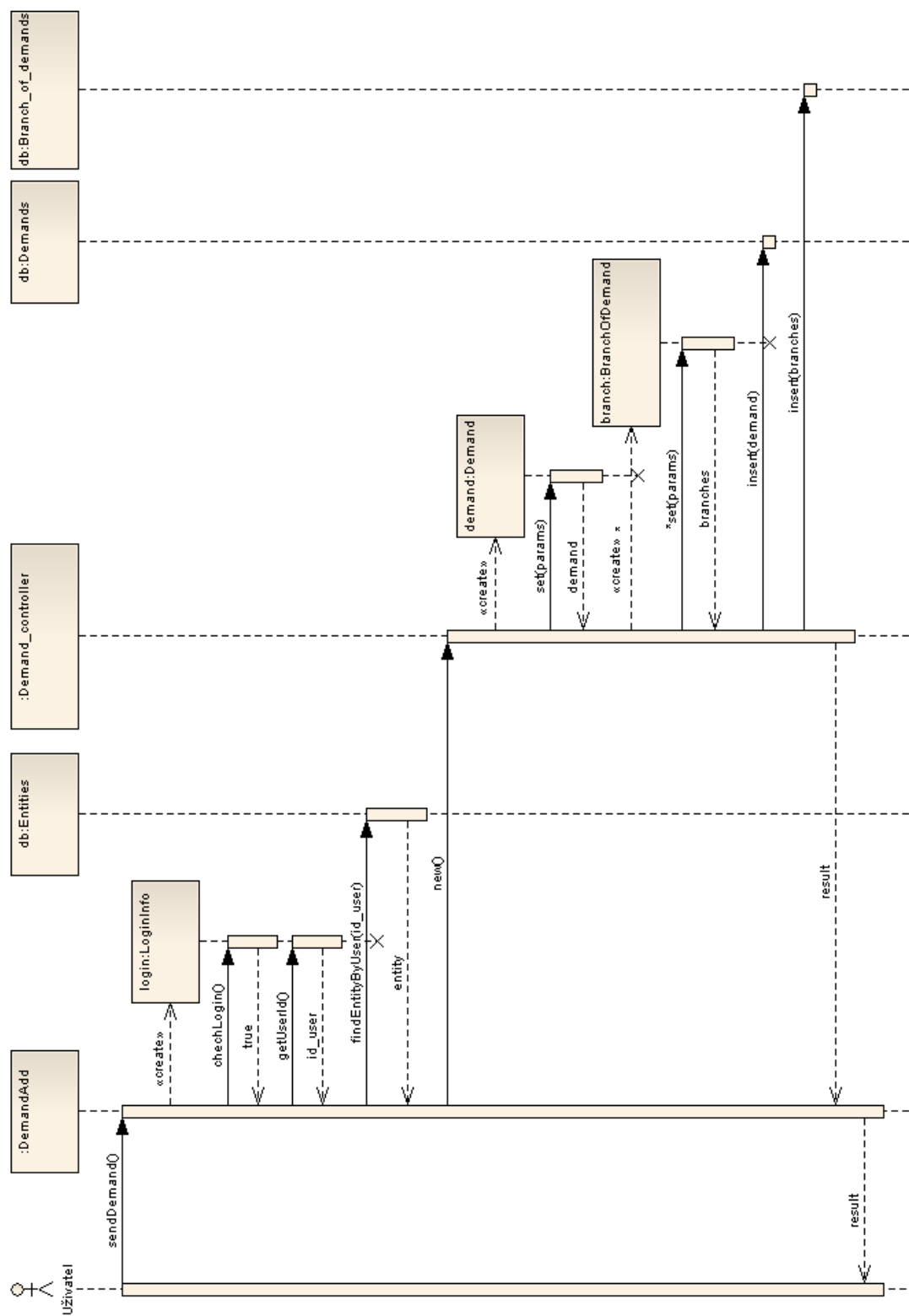
5.2.2 Náhled nasazení

Je vhodné také ukázat, na jakých fyzických prostředcích bude výsledný systém provozován. K tomu nám pomůže diagram nasazení.

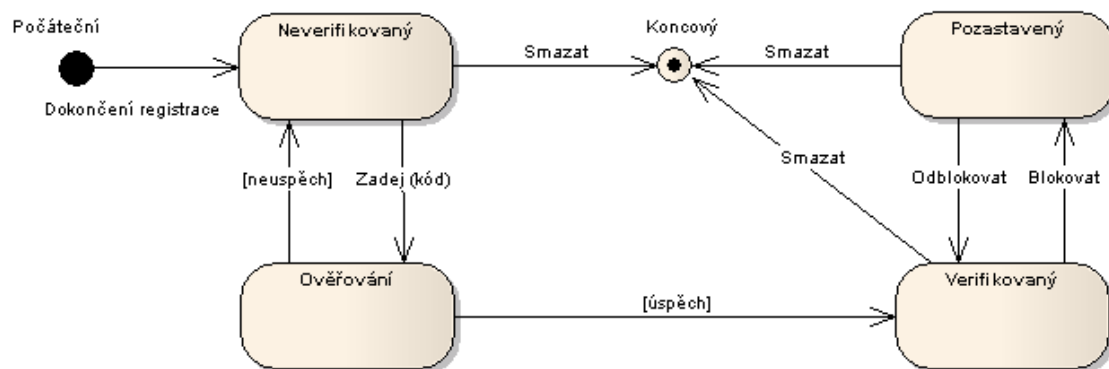
Diagram nasazení popisuje konfiguraci technických prostředků umožňujících běh softwarového systému [2].

Zákazníci mohou k systému přistupovat pomocí webového spojení se serverem. Klientská část webové aplikace by měla běžet pod operačním systémem Windows XP/Vista/Linux/Mac OS. U serverové části je doporučen operační systém Microsoft Windows Server 2008 s podporou ASP.NET 3.5 SP1 a technologií LINQ, AJAX a SILVERLIGHT. Databázový server bude připojen k aplikačnímu serveru prostřednictvím lan. Diagram je zobrazen na obrázku 13.

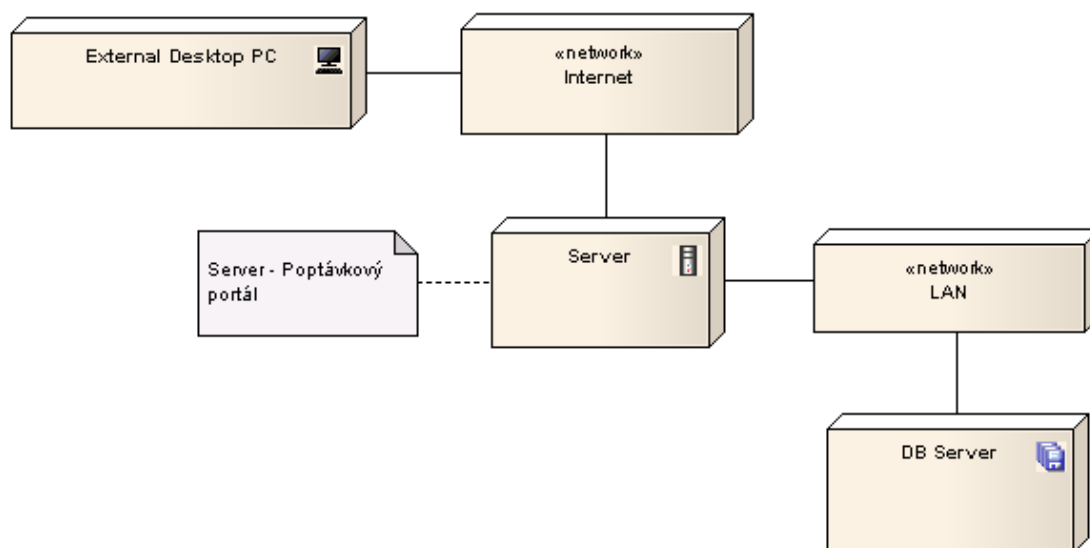
- External Desktop PC - Uživatelské PC umístěné v síti internetu. Uživatelé se pomocí něj mohou pohodlně připojovat do systému za pomoci webového prohlížeče.
- Server - Aplikační server pod operačním systémem Microsoft Windows Server. Přístupný pomocí internetu. Spravuje webový provoz systému.
- DB Server - Poskytuje úložiště dat pro celý systém. Dostupný prostřednictvím lan.



Obrázek 11: Sekvenční diagram – Odeslání poptávky



Obrázek 12: Stavový diagram - Verifikace zákazníka



Obrázek 13: Diagram nasazení

5.2.3 Návrh databáze

Při návrhu databáze bylo snahou navrhnout databázi tak, aby byla konzistentní. Můžou však nastat situace, že bude konzistence databáze porušena i při provozu IS. Proto je snahou, systém bezpodmínečně zajistit i proti možným chybám systému, technickým i programovým.

5.2.3.1 Řešení transakcí Transakce nám dávají možnost provést sadu daných databázových operací jako jeden funkční celek. Musí být tedy nedělitelné (atomické). V praxi to znamená, že navenek se musejí jevit jako jediná operace. Jestliže se transakce dokončí, musejí být zapsány všechny změny, jestliže ne, musejí být všechny změny zrušeny. Při realizaci systému bude využito OR mapperu LINQ to SQL. Díky tomuto nástroji je zajištěno, že při zadávání dotazů je vždy spuštěna transakce. Problém ale nastává, pokud budeme potřebovat spustit více dotazů za sebou a v průběhu vykonávání by došlo k nějaké chybě. Mohlo by se stát, že některá data budou upravena a jiná nikoliv. Došlo by k nekonzistenci dat v databázi. Proto je dobré určit, kde transakce začíná a končí a v případě nějaké chyby, vrátit databázi do původního stavu. Pokud bude při modifikaci dat využita více než jedna nezávislá operace, bude tedy použito transakčních nástrojů, které ASP.NET poskytuje prostřednictvím knihovny *System.Transactions*, která obsahuje funkce pro řešení transakcí. Budou tedy plně využity tam, kde by mohlo dojít k vážnému porušení konzistenci dat v databázi.

5.2.4 Návrh uživatelského rozhraní

Při vývoji webových aplikací je nutný návrh uživatelského rozhraní. Požadavkem je, aby byl vzhled portálu profesionální a rozmístění prvků logické. Zároveň by mělo být uživatelské rozhraní jednoduché a přehledné a důraz bude kladen také na snadnou použitelnost. Šířka stránky bude optimalizována na rozlišení 1024x768 pixelů. Tato minimální šířka by měla vyhovovat většině návštěvníků našeho portálu.

Na základě požadavků na uživatelské rozhraní byly vytvořeny grafické návrhy. Ty jsou součástí přílohy A. Návrhy budou v implementační části kódovány do XHTML a CSS šablon, které budou postupně nasazovány na kostru systému.

6 Implementace

6.1 Architektura systému

Architektura celého systému byla výrazně ovlivněna použitím LINQ to SQL. LINQ samotný obsahuje totiž definici business objektů a definici mapování na tabulky databáze, jedná se prakticky o část datové vrstvy (Data Access Layer). Proto bude datový kontext LINQ to SQL, entity třídy a logika dotazování také umístěna v datové vrstvě.

V praxi by to vypadalo takto: z prezentační vrstvy jsou volány metody umístěné v datové vrstvě, které provedou dotaz, či uloženou proceduru, které jsou mapovány na metody datového kontextu (DataContext). Výsledná data jsou vrácena zpět ke klientovi.

Na obrázku 14 je zobrazena architektura systému.

6.1.1 Prezentační vrstva

Prezentační vrstva má na starost uživatelské rozhraní aplikace. Do této vrstvy patří ve Web Forms webové ovládací prvky a code-behind. Vrstva je zodpovědná za poskytování informací koncovému uživateli a aplikaci. Protože pracujeme s .NET technologií můžeme do této vrstvy zařadit následující typy souborů: .aspx, .ascx, .asax, .cs (code-behind), web.config, .xml, .js, .resx, .css a další. Všechny tyto soubory společně tvoří prezentační logiku a poskytují uživatelské rozhraní pro uživatele.

6.1.2 Datová vrstva

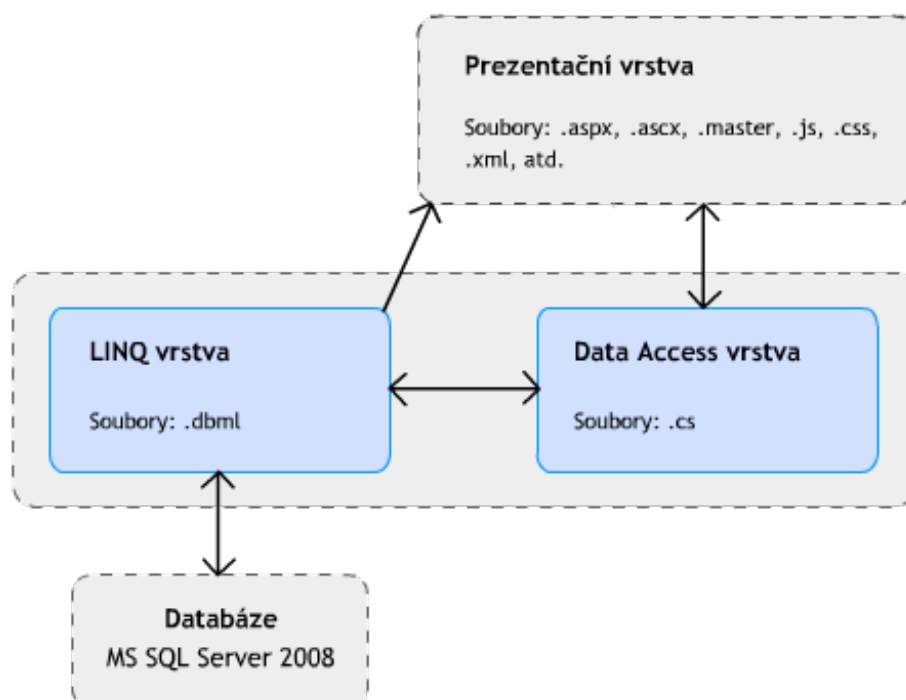
Tato vrstva obsahuje kód pro komunikaci s vrstvou LINQ. V podstatě obsahuje všechny podstatné metody komunikace s vyšší vrstvou a provádí databázové operace. Ukázka takové metody je vidět na zdrojovém kódu níže. Tato metoda má vyhledat všechny zákazníky, kteří jsou zapsáni pod určitou kategorií. Metoda proto přijímá parametr *categoryID* (ID kategorie), vyhledá pomocí LINQ to SQL všechny záznamy v tabulce *Entity*, které mají atribut *CategoryID* roven požadovanému ID a výsledek vrátí jako návratovou hodnotu metody.

```
public IEnumerable<Entity> GetEntitiesByCategory(int categoryID)
{
    DatabaseDataContext db = new DatabaseDataContext();

    IEnumerable<Entity> resultQuery = from p in db.Entities
                                     where p.CategoryID == categoryID
                                     select p;

    return resultQuery.AsEnumerable();
}
```

Výpis 7: Příklad implementace metody v datové vrstvě



Obrázek 14: Architektura systému

6.2 Bezpečnost aplikace

U webových aplikací je bezpečnost celé aplikace prioritní záležitostí. V dnešní době existuje řada útoků, které se snaží systém napadnout. Napadení se může objevit na několika úrovních: od operačního systému serveru až po prohlížeč uživatele. Je tedy třeba ošetřit nejrizikovější místa tak, aby byla aplikace ochráněna.

6.2.1 SQL injection

Tento pojem je na internetu velice rozšířený. Jedná se o podvržení vstupních dat z formulářů nebo jiných vstupů s cílem pozměnit výsledek SQL dotazu. Na internetu existuje stále velké množství webů, které jsou spravovány méně zkušenými programátory, kteří tomuto druhu útoku, buď nepřikládají velký význam a proto jej neřeší, nebo o něm mnohdy ani neví. Přitom rizika jsou veliká. Přinejmenším se útočník dostane tam, kam nemá, ale neprovede nic se samotnou aplikací. V horším případě se může dostat k uživatelským heslům a v nejhorším případě může smazat tabulku nebo pozměnit jejich obsah. Proto je zásadou při psaní každé webové aplikace kontrola a zabezpečení všech vstupních polí od uživatele.

K přístupu k datům v databázi využíváme LINQ to SQL. Můžeme si tedy položit otázku: jak je chráněn LINQ to SQL proti útokům typu SQL injection? SQL injection představuje riziko pro tradiční SQL dotazy, které jsou tvořeny zřetěženými vstupy uživatele.

LINQ to SQL se tomuto druhu útoku brání použitím třídy *SqlParameter*, která reprezentuje parametr v dotazu. Uživatelský vstup je změněn na hodnoty parametru. Tento přístup zabráňuje škodlivým příkazům, které by mohli být použity na uživatelském vstupu.

6.2.2 HTTPS

HTTPS nabízí zabezpečený přístup na webové stránky. Využijeme jej tehdy, pokud chceme zabezpečit, aby informace, které si klient s aplikací vymění, nemohly uniknout. Protože zákaznická sekce systému obsahuje důvěrné informace, použijeme právě HTTPS. Tím budou přenášená data chráněná před odposloucháváním, či podvržením a bude umožněno také ověřit identitu protistrany. HTTPS používá protokol HTTP, data jsou šifrována pomocí SSL. SSL je tedy vrstva/protokol, který zabezpečuje data na přechodu mezi aplikační a transportní vrstvou.

6.2.3 Cross Site Scripting

Tento typ útoku spočívá k narušení webových stránek, kdy jsou zneužity neošetřené výstupy. Proti tomuto útoku se lze bránit důsledným ošetřováním všech řetězců. To znamená, že se budeme snažit, aby nebyl veškerý výstup do HTML stránky zapsán přímo, ale jen jako text. Jinými slovy, musíme nahradit HTML klíčové znaky za jejich entitní vyjádření, tj. například: < za *<*, > za *>*; a & za *&*.

6.3 SEO - optimalizace pro vyhledávače

SEO je technika vytváření a upravování stránek tak, aby ve výsledku dosahovali co nejlepších pozic ve fulltextovém vyhledávání. Cílem této techniky je nalákat na vlastní stránky co největší množství zákazníků. Dle statistik je známo, že přibývá stále více návštěvníků, kteří se na stránky dostanou právě přes výsledky fulltextových vyhledávačů, jako je Seznam, Google, Centrum apod. Dále také vyplynulo, že uživatel věnuje pozornost pouze několika prvním odkazům ve výsledcích vyhledávačů. Proto je dnes SEO považováno za nepostradatelnou součást tvorby webových aplikací.

SEO přináší hned několik výhod:

- zvýšení návštěvnosti stránek zejména cílených návštěvníků, zákazníků
- větší povědomí o poskytovaných produktech
- není třeba investovat za sponzorované odkazy či reklamní kampaně
- výhoda oproti konkurenci
- kvalitní a validní kód
- investice vynaložená do SEO má rychlou návratnost

6.3.1 Metody SEO

Existuje několik technik pro optimalizování stránek. Tyto techniky můžeme rozdělit do dvou částí. První část se zabývá optimalizací vlastních webových stránek - On Page faktory, kdy je potřeba upravit stránky tak, aby vyhovovaly obecně známým pravidlům, které vyhledávače při indexování stránek zohledňují.

On - Page faktory:

- kvalitní a validní zdrojový kód
- sémantika - správné označování jednotlivých prvků na stránce
- poměr klíčových slov k textu a jejich správný výběr
- použití kvalitních textů
- struktura webu a URL
- použití titulků, meta tagů, nadpisů, alt popisků apod. - vše ve správném pořadí
- mapa stránek

Druhá část se zabývá faktory, které nelze ovlivňovat přímo na stránce. Ty se nazývají Off Page faktory.

Off - Page faktory:

- zpětné odkazy - jejich kvalita a počet
- anchor text zpětných odkazů - viditelný text v hypertextovém odkazu
- Google PageRank, S-Rank

Snahou tedy bude navrhnout stránky tak, aby byly pro relevantní dotazy hodnoceny jako kvalitní a aby byly zobrazovány na nejlepších místech ve výsledcích vyhledávání. K tomu využijeme výše zmíněných poznatků, které budou při implementaci využity.

Stejně jako faktory pro dobrou SEO optimalizaci, existují i doporučení, kterých bychom se měli při optimalizaci vyvarovat. Jsou to následující položky:

- nezahalovat text - například stejnou barvou pozadí
- pro důležitý textový obsah nepoužívat obrázky
- nevytvářet duplicitní obsah
- neplnit uměle stránky klíčovými slovy

6.4 URL rewriting

URL rewriting, neboli přepisování url, je technika, jak u dynamicky generovaných stránek docílit uživatelsky přívětivějších url adres. Tato technika je z dnešního pohledu nepostradatelnou pro webové stránky, zakládajících si na SEO. Proto byla věnována rewritingu patřičná pozornost i u tohoto systému.

URL adresa, představuje ve svém původním určení název serveru a fyzickou cestu k souboru na jeho disku. Dynamické weby již však obsahují méně stránek, data se načítají z databáze a nejdůležitějším parametrem je identifikátor, předávaný v query string [10].

Taková běžná URL adresa může vypadat například takto:

`www.coolpoptavka.cz/poptavka.aspx?id=5`

Z adresy je sice patrné, že se zobrazí stránka poptávky s identifikačním číslem 5, ale není jasné jaká poptávka se za URL skrývá. Proto by bylo jistě lepší využít adresy například:

`www.coolpoptavka.cz/poptavka/poptavam-ladeni-klaviru/`

Z adresy je okamžitě jasné, že se zobrazí poptávka s názvem "*poptávám ladění klavíru*". Způsobů, jakými můžeme rewriting realizovat je spousta. Od URL Rewrite Modul pro IIS 7.0, přes rozličná hotová řešení až po vlastní URL modul. Hlavní výhodou a zároveň též nevýhodou URL Rewrite Modulu pro IIS 7.0 je, že v zásadě nemá nic společného s ASP.NET ani jinou serverovou technologií pro vývoj webových stránek. Tento modul umožňuje specifikovat pravidla pomocí regulárních výrazů a kromě rewritingu umí adresy také směrovat, či blokovat. Avšak pokud využijeme vlastního URL modulu, kdy budeme odchyťovat některou z počátečních událostí při požadavku, typicky *BeginRequest* a v ní změníme adresu, dosáhneme stejného výsledku jako u URL Rewrite Modulu [10].

6.4.1 Vlastní rewrite modul

Protože vyvíjený systém obsahuje složitější strukturu adres, byl nakonec vytvořen vlastní URL modul. Tento modul přijímá hezké URL adresy. Modul tuto adresu analyzuje a vrátí zpět interní adresu. Problémem bylo vymyslet způsob, jakým by mohli být přeneseny parametry požadavku, například ID článku z modulu na vnitřní stránky bez použití query string. Jako dobré řešení pro předávání těchto interních parametrů, se nakonec ukázalo, využití kolekce *Items* aktuálního *HttpContextu*. Tato kolekce slouží jako univerzální úložiště sdílené všemi, kdo se na zpracování požadavku podílejí, všemi HTTP moduly a handlers. Koncepčně je podobná třeba Session nebo Cache, které jsou založeny na principu stringový klíč - objektová hodnota. Na rozdíl od nich ale přetrvává kolekce *Items* pouze po dobu zpracování tohoto jednoho požadavku, ne déle. Při rewritingu tedy můžeme uložit požadované parametry zde a ve stránce je zase načíst. Navíc práce s kolekcí *Items* je opravdu triviální.



Generování hezkých url probíhá následovným způsobem. Pokud vkládáme do systému nový článek, aktualitu, či jinou položku, vezme se její název a pomocí generátoru se vytvoří nový titulek, který odpovídá našim SEO požadavkům. Zde jsou uvedena pravidla, která generátor aplikuje:

- Generátor musí zaručit, že nový titulek, který se bude používat pro URL rewrite bude v systému jedinečný. Proto se kontroluje, zda se v systému již nenachází. Pokud by nový titulek jedinečný nebyl, bude na konec titulku přidáno číslo, které se bude iterovat od jedné do n až do doby, dokud nový titulek nebude v systému opravdu jedinečný. Vše můžeme ukázat na příkladu:

6.5 Tag cloud

Jedná se o prvek, který zobrazuje uživatelem generované tagy, či slova z obsahu webové stránky, a typicky se používá k popisu obsahu webu. Tagy jsou obvykle tvořeny pouze jednoslovnými výrazy, které jsou většinou abecedně seřazeny, a důležitost tagů je znázorněna různou velikostí fontu, barvou, či jinak. Tagy mohou být využity jako odkazy, směřující na obsah, který je asociován s tagem. Poprvé byl tento prvek použit na webovém serveru pro sdílení fotografií Flickr. Od té doby se objevuje stále častěji na web 2.0 projektech. Na obrázku 15 je zobrazena ukázka tag cloud [12].

Použit můžeme tento prvek k lepší orientaci na stránce a zároveň bude plnohodnotným navigačním prvkem. Lze jej také využít i k marketingovým účelům, kdy můžeme změnit důležitost tagu uměle, čímž na něj přitáhneme pozornost uživatele.

Tag cloud lze rozdělit na tři typy, dle toho, jaký zdroj dat využijeme:

1. tagy jsou zobrazovány dle četnosti výskytu každé položky
2. globální tag cloud, kdy jsou četnosti generovány napříč všemi prvky a uživateli
3. obsahuje kategorie, kde velikost určuje počet podkategorií, či počet prvků obsažených v kategorii

Pro implementaci tag cloud byl použit velice jednoduchý vzorec [13], díky kterému získáme poměrný koeficient, v našem případě velikost fontu, vůči maximální velikosti, pro daný tag.

$$s_i = \left[\frac{f_{max} \cdot (t_i - t_{min})}{t_{max} - t_{min}} \right] \quad , \text{pro } t_i < t_{min}; \text{ jinak } s_i = 1$$

- s_i : zobrazovaná velikost
- f_{max} : maximální velikost
- t_i : počet prvků
- t_{min} : minimální hodnota
- t_{max} : maximální hodnota

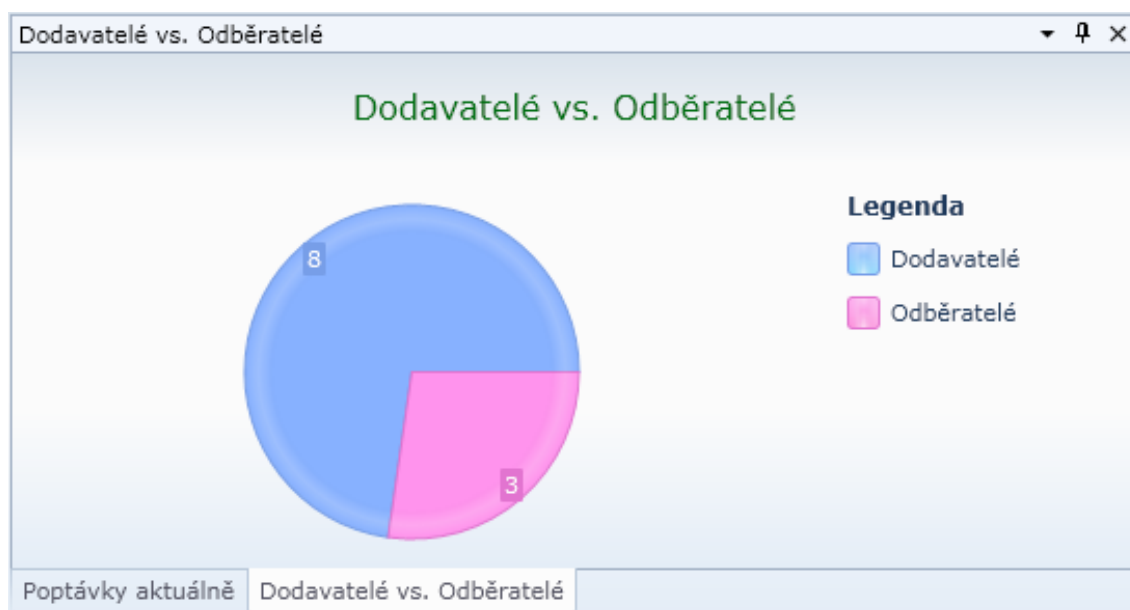
6.6 Silverlight a statistiky

Silverlight dovoluje vytvářet bohaté aplikace plné graficky zpracovaných prvků. Navíc byl od společnosti Telerik použit balíček komponent, které jsou postaveny na Silverlight verzi 3, a které poskytují více jak 40 uživatelských ovládacích prvků. V této práci byly prvky využity pro tvorbu statistik systému.

Primárně byla využita komponenta Chart. Ta nám pomůže transformovat různá data do interaktivních grafů, které umožňují koncovému uživateli analyzovat již kompletní data. Tato komponenta nabízí několik druhů grafů, jako například sloupcové, spojnicové, výsečové grafy a jiné. Můžeme měnit vzhled komponenty, přidávat animace a sama komponenta se také stará o správná měřítka os.

Grafy musí čerpat data z databáze naší aplikace, musíme proto tyto data zajistit a zpřístupnit komponentám. V aplikaci byla vytvořena webová služba, která bude poskytovat potřebná data. Tuto službu budeme u grafů využívat k dolování dat, tedy k tak zvanému data-bindingu. V praxi to vypadá tak, že Silverlight aplikace volá příslušnou webovou metodu z této služby, která vrátí požadovaný výsledek, a ten zobrazíme pomocí komponenty uživateli.

Na obrázku 16 je ukázka grafu, který zobrazuje počet dodavatelů a odběratelů. Stejným způsobem je vytvářena kompletní statistika systému.



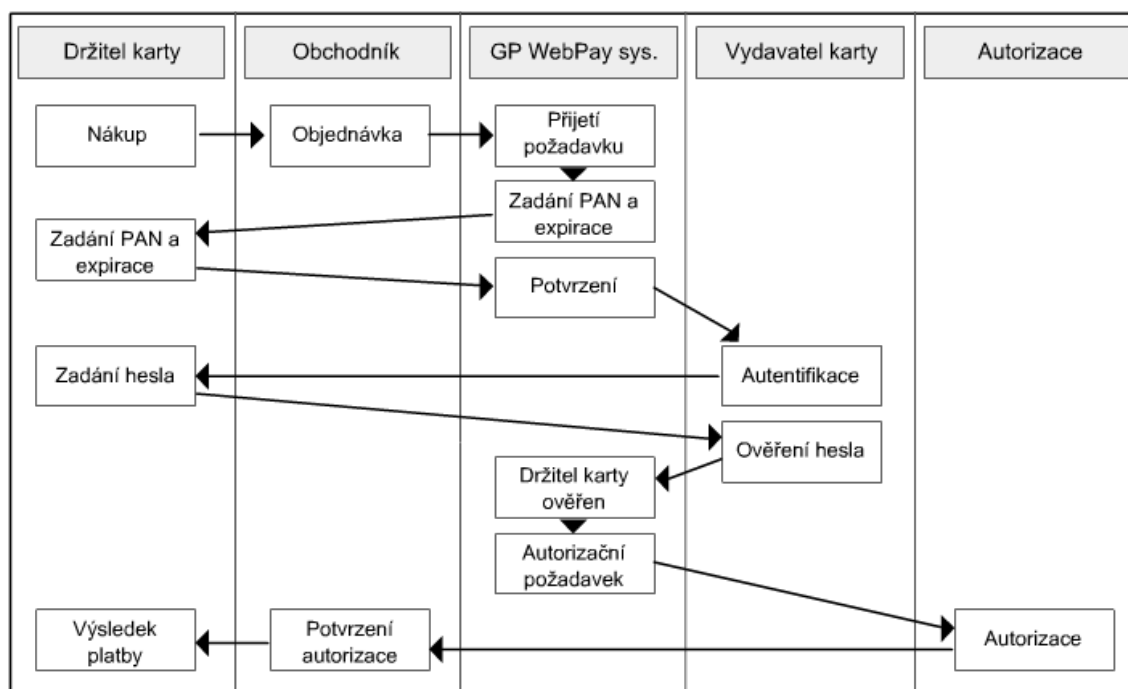
Obrázek 16: Statistika dodavatelů a odběratelů

6.7 Elektronický platební systém

V dnešní době existuje několik elektronických platebních systémů. Všechny mají společný cíl, kterým je přenos určité částky mezi různými účastníky. V našem případě bude využit k platbě za služby, které zákazník bude v systému využívat. Jako hlavní elektronický platební systém byl zvolen GP WebPay (dříve Pay MUZO).

Tato brána umožňuje obchodníkům a prodejcům přijímat platby uskutečněné platebními kartami asociací VISA a MasterCard při on-line transakcích. GP WebPay využívá standard nazvaný 3-D Secure, který poskytuje jedinečnou identifikaci všech stran zapojených do on-line transakce. Díky tomu je vytvořeno úzké spojení mezi vydavatelskou branou a držitelem karty na jedné straně a obchodníkem a účtující bankou na straně druhé. Tento systém navíc používá digitálního podpisu všech dat přenášených mezi platební branou a on-line obchodem. Tím je posílena bezpečnost celého systému. Výhodou je, že tento systém lze implementovat v různých technologiích (PHP, Java, ASP.NET). Komunikace s platební branou je definovaná podle přesně stanovených pravidel [11]. Tyto pravidla obsahuje dokumentace poskytovaná obchodníkům. Mezi parametry a výhody této platební brány patří:

- Podpora všech karet VISA a MasterCard (včetně Maestro a VISA Electron)
- Systém pracuje nepřetržitě v režimu 24x7x365
- Umožňuje non-stop monitoring dostupnosti služeb
- Jsou dodržovány nejpřísnější bezpečnostní standardy (3-D Secure)



Obrázek 17: GP WebPay - 3-D Secure

- Certifikace prostřednictvím společností, které vydávají karty

6.7.1 3-D Secure

Vzhledem k možnosti snadného zneužití plateb prostřednictvím platebních karet v prostředí internetu podporuje GP WebPay standard, který je kontrolován příslušnou asociací VISA nebo MasterCard.

Při přijetí požadavku na provedení platby platební kartou se kontroluje, zda je pro kartu požadována autentifikace držitele karty, či nikoliv. V případě, že ano, držitel karty musí potvrdit svoji identitu. Je vyžadována tajná informace, kterou sdílí s vydavatelem karty (většinou PIN kód). Na základě obdržených výsledků se povoluje, či zamítá možnost dalšího zpracování objednávky [11]. Zabezpečení 3-D Secure zachycuje obrázek 17.

6.8 Exporty

Exporty mají uživatelé poskytnout možnost, jak získat potřebný výstup dat v požadovaném formátu. Dle požadavků zadavatele obsahuje poptávkový portál několik tiskových sestav, které bude umožněno vyexportovat. Při implementaci bylo nutné rozhodnout, jaké výstupní formáty bude systém nabízet. Jako primární byl zvolen formát PDF, který je díky své přenositelnosti, nejpoužívanějším formátem při exportu faktur a jiných dokumentů.

Pro ASP.NET existuje několik placených i open-source nástrojů, které umožňují vytvářet exporty. Microsoft poskytuje dokonce řešení Crystal Reports. Bohužel podpora toho nástroje končí s uvedením nové verze Visual Studia 2010. Žádný z blíže zkoumaných, však neposkytuje takovou škálu funkcí a výhod, jako nástroj od společnosti Telerik, *Telerik Reporting*.

Tento nástroj je vytvořen přímo pro .NET aplikace. Je možné jej použít v aplikacích Windows Form, WebForms a dokonce i v aplikacích psaných v technologii Silverlight. Telerik Reporting podporuje exporty dat do všech populárních formátů, jako jsou PDF, Excel, RTF, TIFF, atd. Navíc umožňuje pracovat s exporty v design modu a nabízí kompletní API pro Visual Studio 2005/2008/2010. Díky snadnému propojení s naším systémem můžeme také vytvářet exporty, které budou využívat dat z databáze.

7 Závěr

V úvodu této diplomové práce byla zmíněna náplň a cíle, kterých by mělo být dosaženo. Chtěl bych proto celý proces vývoje shrnout a nakonec přidat zhodnocení.

První část práce je zaměřena na použité technologie, kde byly shrnuty všechny, které byly při vývoje práce použity. S každou technologií jsem se proto musel seznámit a následně ji použít při implementaci systému.

Dále bylo nutné objasnit základní pojmy, které se vyskytují v oblasti poptávkových portálů a analyzovat již existující poptávkové systémy, jejich funkčnost, nastavení pracovních procesů a administrativní nutnosti systému. Detailně muselo být prozkoumáno hned několik systémů. V této práci byly pak uvedeny ty nejdůležitější. Na základě této analýzy bylo zjištěno, jakým způsobem dostupné systémy pracují a jaké funkce poskytují. Díky těmto poznatkům byla sepsána předběžná představa zadavatele a inovace, které by měl nově vyvíjený systém umět oproti konkurenčním systémům. Všechny informace byly následně použity při specifikování požadavků na systém.

Požadavky na systém byly zpracovány ve spolupráci se zadavatelem a do specifikace byly zahrnuty i ty funkce systému, které budou realizovány až v dalších verzích systému. Sběr informací probíhal prostřednictvím častých schůzek a konzultací se zadavatelem. Další část práce je věnována analýze a návrhu. V této fázi byly navrženy části systému, včetně návrhu databáze. Ten hraje při vývoji webových aplikací tohoto rozsahu velmi důležitou roli.

Důraz byl kladen také na poslední fázi a to implementaci systému. Ta zabrala podstatnou část vývoje této diplomové práce.

Výsledkem je kompletní poptávkový portál schopný komerčního provozu. Požadavky, které na tento systém byly kladeny, byly splněny. Nad rámec práce byl vyvinut modul redakčního systému, který lze použít i samostatně, či jako podpůrný modul do jiného projektu. Také některé z požadavků, které měli být naimplementovány až v dalších verzích systému, se podařilo začlenit již do této práce. Jedná se především o platební systém pomocí elektronických karet.

7.1 Další vývoj projektu

Při vývoji této práce vyvstalo několik nových námětů a možných rozšíření, které bychom mohli do systému zahrnout. Jedná se například o rozšíření modulu pro hromadné zasílání emailů. Bylo by totiž možné, vytvořit v administraci propracovaný šablonový systém pro emaily, s možností nastavení hlavičky, či patičky, nebo i těla emailu. Také by bylo možné přiřazovat emailům různé akce. Například kdy mají být odeslány, zda se jedná o hromadný email, atd.

Další možností je rozšíření systému o další platební systémy. Poté by si uživatel mohl vybrat ten způsob platby, který preferuje. Jednalo by se například o systém PayPal, který je oblíbený zejména na zahraničních serverech jako je e-Bay.

Zajímavou možností pro poptávkový portál by bylo také vytvořit moduly umožňující dodavatelům nabízet, či inzerovat své akční nabídky na viditelném místě na portálu. Tato služba by mohla být dalším zdrojem financí pro zadavatele.

Tento systém je naimplementován tak, aby bylo snadné jej měnit, nebo přidávat další moduly. Je tedy možné zamyslet se nad možností prodeje licencí tohoto systému. Muselo by se však promyslet, zda by tento krok nevedl k příliš velké konkurenci na trhu poptávkových portálů.

8 Literatura

- [1] MACDONALD, Matthew; SZPUSZTA, Mario. *ASP.NET 3.5 a C# 2008 : tvorba dynamických stránek PROFESIONÁLNĚ*. Vydání první. Brno : Zoner Press, 2008. 1531 s. ISBN 978-80-7413-008-3.
- [2] VONDRÁK, Ivo. *Úvod do softwarového inženýrství*. Verze 1.1. Ostrava : VŠB - Technická univerzita Ostrava, 2002.
- [3] ARLOW, Jim; NEUSTADT, Ila. *UML 2 a unifikovaný proces vývoje aplikací ; Objektově orientovaná analýza a návrh prakticky*. Vydání první. Brno : Computer Press, 2007. 576 s. ISBN 978-80-251-1503-9
- [4] KANISOVÁ, Hana; MÜLLER, Miroslav. *UML srozumitelně*. Vydání první. Brno : Computer Press, 2004. 158 s. ISBN 80-251-0231-9.
- [5] MSDN [online]. [cit. 2009-11-02]. Using LINQ with APS.NET. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/library/bb907622.aspx>>.
- [6] Interval.cz [online]. 03.11.2008 [cit. 2010-1-18]. Silverlight - co je Silverlight?. Dostupné z WWW: <<http://interval.cz/clanky/silverlight-co-je-silverlight/>>.
- [7] EPoptávka.cz [online]. [cit. 2009-05-11]. EPoptávka.cz - poptávky a veřejné zakázky. Dostupné z WWW: <<http://www.epoptavka.cz/>>.
- [8] AAA POPTÁVKA.CZ [online]. [cit. 2009-05-11]. AAA POPTÁVKA.CZ - Poptávky, veřejné zakázky a katalog firem. Dostupné z WWW: <www.aapoptavka.cz>.
- [9] NiceOne [online]. [cit. 2009-05-11]. NiceOne, poptávkový portál. Dostupné z WWW: <www.niceone.cz>.
- [10] VALÁŠEK, Michal. ASPNET.CZ [online]. [cit.2010-03-08]. URL rewriting v IIS a ASP.NET. Dostupné z WWW: <<http://www.aspnet.cz/Articles/218-url-rewriting-v-iis-a-asp-net.aspx>>.
- [11] Web Revolution [online]. [cit. 2010-03-14]. Pay MUZO. Dostupné z WWW: <<http://www.web-revolution.cz/paymuzo>>.
- [12] FRIEDMAN, Vitaly. *Smashing Magazine* [online]. 2007-11-7 [cit. 2010-02-23]. Tag Clouds Gallery: Examples And Good Practices. Dostupné z WWW: <<http://www.smashingmagazine.com/2007/11/07/tag-clouds-gallery-examples-and-good-practices/>>.
- [13] Tag Cloud [online]. [cit. 2010-02-23]. Tag cloud - Wikipedia, the free encyclopedia. Dostupné z WWW: <http://en.wikipedia.org/wiki/Tag_cloud>.

A Uživatelské rozhraní a ukázky aplikace

V této části jsou ukázky uživatelského rozhraní nového poptávkového portálu a jsou zde popsány některé uživatelské prvky, které se vyskytují v aplikaci.

A.1 Uživatelské rozhraní

Na obrázku 18 vidíme návrh uživatelského rozhraní. Tento návrh vznikl na základě specifikace požadavků a představuje základní vzhled aplikace. Na stránce se nachází několik prvků. V záhlaví je logo *COOL poptávka.cz*, pod kterým bude poptávkový portál vystupovat, a pole pro fulltextové hledání na portálu. Následuje hlavní menu, které můžeme editovat v administraci. Stránka se dále dělí na levý a pravý panel. V levém panelu se zobrazuje obsah aktuální stránky. Zde se mohou objevovat prvky jako je katalog firem, přehled poptávek, poslední vložené poptávky, poslední registrovaní uživatelé a jiné. Pravý panel obsahuje doplňující prvky jako je přihlašovací formulář. Tento využívají odběratelé a dodavatelé k přihlašování ke svému uživatelskému účtu. Dále se tu nachází moduly redakčního systému Aktuality a Novinky, pod nimi je umístěn tagcloud (neboli mračno tagů - viz. kapitola 6) a následně je zde vyhrazeno místo pro reklamní bannery. Stránku uzavírá spodní menu, které je také možné upravovat v administraci.

A.2 Ukázky aplikace

Nyní probereme základní práci s některými prvky systému.


A.2.1 Prezentace dat

Pro prezentaci dat z databáze je využíváno komponenty RadGrid od společnosti Telerik. Tato komponenta je určena pro ASP.NET AJAX a díky AJAX se jeví jako desktopová komponenta. Vyznačuje se také vysokým výkonem při práci s daty. Na obrázku 19 tato komponenta zobrazuje všechny zákazníky v naší aplikaci. Obrázek je označen čísly, které označují jednotlivé části této komponenty. Část označená čísly 1 a 5 je určena pro stránkování dat. Je dovoleno stránkovat po jednom listu, nebo můžeme přeskokovat z prvního listu na poslední, či dokonce změnit počet záznamů na stránce.

Číslo 2 ukazuje lištu, která obsahuje jednotlivé atributy (ekvivalent atributu databázové tabulky) zobrazovaných dat. V našem příkladu to jsou atributy Firma/Jméno, Typ, Kód, Registrováno a Verifikace. Tato lišta nemá pouze informativní charakter. Můžeme díky ní data třídit. Stačí kliknout na kterýkoliv atribut a data se budou třídit sestupně podle něj. Pokud na něj klikneme podruhé, data se setřídí vzestupně. Můžeme tedy dostávat různě setříděné seznamy, dle našich aktuálních požadavků.

V části 3 můžeme ve výpisu dat i vyhledávat. V našem příkladu vidíme dvě pole, pro vyhledávání podle atributu Firma/Jméno, či Kód. Vše funguje díky AJAXu naprosto plynule.

Část 4 obsahuje jednotlivé záznamy zákazníků. Nad každým záznamem je možné provádět doplňující operace. Na obrázku jsou vidět tři ikony. První (obálka) představuje



HLEDEJ

POPTÁVÁM
NABÍZÍM
ÚVOD
OBCHODNÍ PODMÍNKY
PŘIHLÁŠENÍ
REGISTRACE
KONTAKTY

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

AUTO - MOTO
Náhradní díly (42),
Nákladní auta (6), Osobní

JAK REGISTROVAT

VLOŽIT FIRMU ZDARMA

LOGIN

HESLO

Zapamatovat heslo

PŘIHLÁSIT SE

AKTUALITY

15.3.

1. Vkládejte poptávky na cokoliv a čekejte na vhodné dodavatele.
2. Hledejte v již vložených poptávkách a kontaktujte poptávající.
3. Registrujte se jako dodavatel a čekejte na zaslané poptávky.

15.3.

Novinky v registraci - Registrujte se jako dodavatel a čekejte na zaslané poptávky.

POSLEDNÍ POPTÁVKY

POSLEDNÍ NABÍDKY

Práce v IT na částečný úvazek - Moravskoslezský kraj

1.3.2010

Práce v IT na částečný úvazek - Moravskoslezský kraj

1.3.2010

Práce v IT na částečný úvazek - Moravskoslezský kraj

1.3.2010

Práce v IT na částečný úvazek - Moravskoslezský kraj

1.3.2010

NAPOSLEDY REGISTROVANÍ DODAVATELÉ

SKZ NORD s.r.o.

dnes

TJ Slavov Stará Boleslav

dnes

Marek Jiřík

dnes

Miroslav Čermák

dnes

OMEGASTAVOS s.r.o.

včera

Tomáš Zvoníček

včera

Petr KEŠNER

včera

Martin Franěk

včera

Tomáš Zvoníček

včera

Martina Krejsarová

10.3.2010

AR Střechy-Roman Fajkus

10.3.2010

Jiří Vacula

10.3.2010

Radek Michalica

10.3.2010

SKZ NORD s.r.o.

dnes

TJ Slavov Stará Boleslav

dnes

Marek Jiřík

dnes

Miroslav Čermák

dnes

OMEGASTAVOS s.r.o.

včera

Tomáš Zvoníček

včera

Petr KEŠNER

včera

Martin Franěk

včera

Tomáš Zvoníček

včera

Martina Krejsarová

10.3.2010

AR Střechy-Roman Fajkus

10.3.2010

Jiří Vacula

10.3.2010

Radek Michalica

10.3.2010

POPTÁVKY ZDARMA

Poptávka - Dodávka a montáž dřevěné terasy s pergolou k RD

dnes

Poptávka - Hlídní dětí

dnes

Poptávka - Investor nebo půjčka na rozjezd obchodu Maso-Uzeniny.

dnes

Poptávka - Dodavatel zahradní techniky

dnes

Poptávka - Termostatická vanová baterie

včera

Poptávka - Nesámovaná prkna Douglaska

včera

Poptávka - Obchodní zástupce v potravinářství

včera

Poptávka - Dodavatel na gastro doplňky a koupelnové doplňky

včera

Poptávka - Termostatická vanová baterie

včera

Poptávka - Nesámovaná prkna Douglaska

10.3.2010

Poptávka - Obchodní zástupce v potravinářství

10.3.2010

Poptávka - Dodavatel na gastro doplňky a koupelnové doplňky

10.3.2010

Poptávka - Obchodní zástupce v potravinářství

10.3.2010

starší poptávky >

POPTÁVÁM AUTO - MOTO

CESTOVÁNÍ OBCHODNÍ

PODMÍNKY KONTAKTY

SLEVY AKCE KATEGORIE

AKUALITY

BANNER

ÚVOD

OBCHODNÍ PODMÍNKY

PŘIHLÁŠENÍ

REGISTRACE

KONTAKTY

info@coolpoptavka.cz

info: +420 111 111 111

Obrázek 18: Uživatelské rozhraní

Firma/Jméno	Typ	Kód	Registrováno	Verifikace
Bontom s.r.o.	dodavatel	o000016	20.4.2010 14:41:13	Verifikovaný
Jose Group	dodavatel	o000015	20.4.2010 12:51:01	Verifikovaný
Jose Group	dodavatel	o000014	20.4.2010 12:45:14	Čekající
Podravka s.r.o.	dodavatel	o000013	19.4.2010 18:33:00	Čekající
dasdasda	dodavatel	o000011	14.4.2010 14:43:28	Čekající
Xien Market	dodavatel	o000010	14.4.2010 13:28:02	Čekající
Fiala Lukáš	odběratel	o000009	13.4.2010 12:15:05	Verifikovaný
Jedlička Karel	odběratel	o000008	13.4.2010 12:03:40	Verifikovaný
PepaNos a.s	dodavatel	o000007	13.4.2010 11:24:30	Verifikovaný
Lukas S	dodavatel	o000006	12.4.2010 18:32:20	Čekající

Obrázek 19: Prezentace dat pomocí komponenty RadGrid

funkci odeslání interní zprávy zákazníkovi, druhá slouží pro editaci záznamu a třetí pro smazání záznamu.

A.2.2 Interní zasílání zpráv

Pomocí interních zpráv mohou mezi sebou komunikovat uživatelé systému. Zprávy může zasílat dodavatel, odběratel, administrátor nebo sám systém. Například dodavatel může pomocí interní zprávy odeslat nabídku na poptávku odběratele, administrátor může zasílat různé informace zákazníkům a systém může zasílat hromadná upozornění, vystavené faktury, atd.

Na obrázku 20 je ukázka okna pro zasílání interní zprávy, které obsahuje několik údajů. Jako první údaj je příjemce, kterému bude zpráva zaslána, a dále předmět zprávy. K interní zprávě také můžeme připojit přílohu, která je pro zákazníky omezena na tři soubory. Administrátor i systém velikost přílohy omezenu nemají. Následuje pole pro text zprávy. Zprávu můžeme následně odeslat nebo zrušit.

Princip interních zpráv je velice podobný zasílání e-mailů. Proto je pro uživatele používání této funkce naprosto intuitivní.

A.2.3 Tisková sestava

Systém umožňuje získat potřebný výstup dat v požadovaném formátu. V systému jsou tedy tiskové sestavy pro poptávky, nabídky, výzvy k platbě, faktury a další. Tyto sestavy lze exportovat do různých formátů, jako jsou PDF, Excel a jiné, nebo je můžeme přímo tisknout. Na obrázku 21 je ukázka vyexportované faktury do formátu PDF.




Odpověď:

Komu: Podravka s.r.o.

Předmět: Nějaký předmět



☐ file.pdf

Příloha:

B   

Text, který chceme poslat pomocí interní zprávy

Text:

Obrázek 20: Okno pro odeslání interní zprávy

Faktura

2010000001

Odběratel:

Bontom s.r.o.

Beníkova 88

828 82 Okurkov

IČ: 82883881

DIČ:

Korespondenční adresa:

Bontom s.r.o.

Beníkova 88

828 82 Okurkov

Dodavatel:

XEVOS Solution s.r.o

Vědecko-technologický park Ostrava

708 00 Ostrava - Pustkovec

IČ: 27831345

DIČ: CZ27831345

Bankovní spojení: M-Bank**Číslo účtu: 1122334455/6210**

IBAN: 0

Detail platby:

Variabilní symbol: 20800000002

Konstatní symbol: 0

Datum vystavení: 20.4.2010

Datum uskutečnění zdan. plnění: 20.4.2010

Datum přijetí platby:

Datum splatnosti: 20.4.2010

Popis položky

Cena

Tarif: Na zkoušku

300,00 Kč

Zvýraznění

500,00 Kč

[UHRAZENO]

Cena bez DPH: 40,00 Kč

Celkem s DPH: 800,00 Kč

CELKEM K ÚHRADĚ: 800,00 Kč

Obrázek 21: Export faktury ve formátu PDF

B Datový slovník

Tato část obsahuje datový slovník redakčního systému.

CMS.actuality – tabulka aktualit

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
title	nvarchar(100)	N	N	N	název aktuality
description	nvarchar(2000)	N	Y	N	popis
text	nvarchar(MAX)	N	Y	N	text
seo_keyword	nvarchar(200)	N	Y	N	SEO klíčová slova
seo_description	nvarchar(200)	N	Y	N	SEO popis
is_actuality	bit	N	N	N	0 - novinka, 1 - aktualita
rewrite_url	nvarchar(100)	N	N	N	název pro url
date_published	datetime	N	N	N	datum publikování
date_added	datetime	N	N	N	datum vytvoření
is_publish	bit	N	Y	N	je publikován

Tabulka 16: Datový slovník - CMS.actuality

CMS.banner – tabulka bannerů

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
title	nvarchar(100)	N	N	N	název
image_url	nvarchar(100)	N	N	N	název souboru
navigate_url	nvarchar(100)	N	N	N	adresa odkazu
alternate_text	nvarchar(100)	N	Y	N	alternativní text
width	int	N	Y	N	šířka banneru
height	int	N	Y	N	výška banneru
is_visible	bit	N	N	N	je viditelný

Tabulka 17: Datový slovník - CMS.banner

CMS_article – tabulka článků

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
id_category	int	FK	Y	Y	id kategorie
title	nvarchar(100)	N	N	N	název článku
description	nvarchar(2000)	N	Y	N	popis
text	nvarchar(MAX)	N	Y	N	text
date_publish	datetime	N	N	N	datum publikování
is_front_page	bit	N	N	N	je na hlavní straně
hits	int	N	N	N	počet zobrazení
author	nvarchar(100)	N	Y	N	autor článku
seo_keyword	nvarchar(200)	N	Y	N	SEO klíčová slova
seo_description	nvarchar(200)	N	Y	N	SEO popis
rewrite_url	nvarchar(100)	N	N	N	název pro url
is_published	bit	N	N	N	je publikován
is_trash	bit	N	N	N	je smazán

Tabulka 18: Datový slovník - CMS_article

CMS_category – tabulka kategorií CMS

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
id_section	FK	N	N	N	id sekce
title	nvarchar(100)	N	N	N	název kategorie
description	nvarchar(1000)	N	Y	N	popis
date_added	datetime	N	N	N	datum vložení
is_published	bit	N	N	N	je publikován

Tabulka 19: Datový slovník - CMS_category

CMS_section – tabulka sekcí CMS

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
title	nvarchar(100)	N	N	N	název sekce
description	nvarchar(1000)	N	Y	N	popis
date_added	datetime	N	N	N	datum vložení
is_published	bit	N	N	N	je publikován

Tabulka 20: Datový slovník - CMS_section

CMS_menu – tabulka menu CMS

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
title	nvarchar(100)	N	N	N	název menu
description	nvarchar(500)	N	Y	N	popis

Tabulka 21: Datový slovník - CMS_menu

CMS_menu_item – tabulka menu položek

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
id_parent_menu_item	int	FK	Y	Y	id rodičovské položky
id_menu	int	FK	N	Y	id menu
id_type	int	FK	N	Y	id typu
id_entity	int	FK	Y	N	id objektu
name	nvarchar(100)	N	N	N	název položky
rewrite_url	nvarchar(120)	N	N	N	název pro url
link	nvarchar(100)	N	Y	N	url odkazu
rank	int	N	N	N	pořadí položky
is_published	bit	N	N	N	je publikovaná

Tabulka 22: Datový slovník - CMS_menu_item

CMS.type – tabulka typů CMS

Atribut	Datový typ	Klíč	Null	Index	Popis
id	int	PK	N	Y	id
code	nvarchar(50)	N	N	N	kód typu
name	nvarchar(50)	N	Y	N	název typu
description	nvarchar(500)	N	Y	N	popis

Tabulka 23: Datový slovník - CMS.type